

# Numerical methods for two person games arising from transboundary pollution with emission permit trading<sup>☆</sup>



Junyu Lai<sup>a</sup>, Justin W.L. Wan<sup>a</sup>, Shuhua Zhang<sup>b,\*</sup>

<sup>a</sup> Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada

<sup>b</sup> Coordinated Innovation Center for Computable Modeling in Management Science, Tianjin University of Finance and Economics, Tianjin 300222, China

## ARTICLE INFO

### MSC:

65M06  
65M12  
65M22  
91B76

### Keywords:

Transboundary pollution  
Differential games  
Emission controls  
HJB equation  
Finite difference method  
Wide stencil

## ABSTRACT

We propose finite difference methods for solving two dimensional Hamilton–Jacobi–Bellman (HJB) equations and systems arising from the modelling of transboundary pollution with emission permits trading. We prove that our numerical scheme for the HJB equations from cooperative game is consistent, stable, and monotone, therefore guarantees convergence to viscosity solutions. To ensure our scheme is fully implicit and unconditionally monotone, we combine wide stencil with narrow stencil in the discretization. We address coupling between unknown variables and emission controls using policy iteration. We solve the coupled systems of HJB equations from non-cooperative game efficiently through policy-like iteration. We give a theoretical analysis of our scheme for solving the coupled system and find that it converges under certain conditions. Finally, we show that our numerical scheme achieves higher order of convergence than the finite volume method proposed originally.

© 2019 Elsevier Inc. All rights reserved.

## 1. Introduction

Transboundary pollution [1] refers to the situation that the pollution originated from one country or region adversely affects the environment of other countries or regions through crossing the political boundaries. For example, air pollutants like sulphur dioxide (SO<sub>2</sub>), nitrogen oxides (NO<sub>x</sub>) and carbon monoxide (CO) can be carried thousands of kilometers by wind from their sources. Unlike some political problems, most environmental pollution is transboundary, which is difficult to control and can not be dealt by simply building a wall on the borders between neighboring countries. Some international agreements like Kyoto Protocol [2] and US–Canada Air Quality Agreement [3], have been signed to regulate and control transboundary pollutant emission. Modeling transboundary industrial pollution problems can help a country to make decisions on international cooperation and negotiation on environmental matters with the objective of achieving maximum net profits. A country may choose not to cooperate with others if such cooperation is expected to reduce her profits and no compensation is paid.

<sup>☆</sup> This project was supported in part by the Natural Sciences and Engineering Research Council of Canada, the Major Research Plan of the National Natural Science Foundation of China (91430108), the National Basic Research Program of China (2012CB95504), the National Natural Science Foundation of China (11771322) and the Major Program of Tianjin University of Finance and Economics (ZD1302).

\* Corresponding author.

E-mail addresses: [j33lai@uwaterloo.ca](mailto:j33lai@uwaterloo.ca) (J. Lai), [jwlwan@uwaterloo.ca](mailto:jwlwan@uwaterloo.ca) (J.W.L. Wan), [shuhua55@126.com](mailto:shuhua55@126.com) (S. Zhang).

Differential game modeling is treated as an effective method to solve transboundary pollution control problems and to analyze the interactions between the dynamic evolution of the environment and the participants' strategic behaviors. A cooperative differential game of transboundary industrial pollution is presented by Yeung [4], which first derives the time-consistent solutions in the game on pollution control with industries and governments modeled as separate entities. After that, Yeung and Petrosyn [5] present a cooperative stochastic differential game of transboundary industrial pollution where two novel features are provided. The first one is that industrial production creates short-term local and long-term global negative impacts on the environment. In addition to that, they derive a subgame consistent cooperative solution in the differential game along with a payment distribution mechanism. Following the analytical framework of Yeung and Petrosyn [4,5], Li [6] models the differential game of transboundary industrial pollution in which emission permits trading is taken into account as a constant. However, emission permits price has been studied theoretically and empirically and researchers believe that it should contain a stochastic element [7–10]. Following the work of Li [6], Chang et al. [11] generalize emission permits price to follow a geometric Brownian motion (GBM). Through classical stochastic analysis, Chang et al. [11] derive an Hamilton–Jacobi–Bellman (HJB) partial differential equation of two dimensions (pollution stock and permits price) for the cooperative games and a system of two HJB 2D equations for the non-cooperative games. They choose finite volume method (FVM) to solve the equations and obtain convergence of the order  $O(h^{0.6353})$ . Wang et al. [12] solve HJB equations arising from optimal feedback control problems numerically using an explicit and conditional stable finite difference scheme. Jensen et al.'s [13] work explore the convergence of finite element methods (FEMs) for HJB equations with respect to optimal control problems. In this paper, we consider finite difference method (FDM) and aim at achieving higher convergence order for viscosity solutions [14] to the PDE for cooperative game.

The sufficient condition for a numerical scheme to converge to viscosity solution is that the scheme is consistent, stable and monotone [15]. Pooley et al. [16] show, for the one dimensional case, that seemingly reasonable discretizations of the stochastic PDE may not converge to the viscosity solution, which is the financially relevant solution. For 2D HJB equations, the presence of the cross derivative term makes the construction of monotone scheme non-trivial. Pooley et al. [17] explore two factor uncertain volatility models but the scheme they provide does not guarantee monotonicity which is a sufficient condition for computing viscosity solutions. One approach is to rotate the coordinate system to eliminate the cross derivative terms but the orthogonality of the coordinate system is no longer preserved, which could complicate the problem. Clift and Forsyth [18] and Amarala and Wan [19] enforce spacing restrictions on the original finite difference grid to obtain positive coefficient conditions for monotonicity. However, this method may not work if the diffusion tensors are non-constant. Debrabant and Jakobsen [20] propose explicit wide stencil schemes, which achieve first order convergence. Ma and Forsyth [21] implement a fully implicit, consistent, unconditionally monotone numerical scheme for two factor uncertain volatility model using wide stencil. They iterate the coupled problem based on policy and recompute the discretization matrix at each policy iteration. In this paper, we apply the wide stencil to the HJB equation from cooperative games. To improve efficiency, the diffusion parts in the discretization matrix are precomputed before iteration starts. As the wide stencil discretization is more expensive for solving the linear system, we try to use the seven point stencil as much as possible.

We summarize the main results of this paper as follows:

- Develop a fully implicit, consistent, unconditionally  $l_\infty$  stable and monotone numerical scheme for the HJB equation from cooperative games.
- Solve the coupled discretized equations using a new policy iteration.
- A theoretical proof that our numerical scheme converges to viscosity solution.
- Solve the system from the non-cooperative game efficiently by embedding controls into policy iterations, which saves time for recomputing the discretization matrices and solving linear systems.
- Provide an analysis on the convergence of the above embedded policy iteration and show the convergence.
- The numerical results present first order convergence for both cooperative and non-cooperative games.

The rest of this paper is organized as follows. Section 2 introduces the derivation of cooperative and non-cooperative games, and discusses the formulations of HJB equations and system from these differential games. Section 3 focuses on numerical solutions to HJB equations from the cooperative game while Section 4 deals with solving the system of HJB equations for the non-cooperative game. Numerical results are presented in Section 5. We conclude our paper in Section 6.

## 2. The differential games

We briefly introduce the derivation of the differential game framework that involves two countries or regions (see Chang et al. [11] for details). Let  $E_i$  ( $i = 1, 2$ ) denote the emission of region  $i$ , which can be controlled by the corresponding region. As in Chang et al. [11], we relax the restriction of the controls  $E_i$  and make them unbounded. One may refer to Chang et al. [11] for detailed reasons. Assume that the stock of pollution  $P$  and the emission permits price  $S$  are stochastic and follow a geometric Brownian motion (GBM) respectively:

$$dP = (E_1 + E_2 - \theta_p P)dt + \sigma_p P dW_p, \quad (2.1)$$

and

$$dS = \mu_s S dt + \sigma_s S dW_s, \quad (2.2)$$

where  $\theta_p$  is a positive constant representing the exponential decay rate of pollution,  $\mu_S$  denotes the drift rate, which is also a constant,  $\sigma_p > 0$  and  $\sigma_S > 0$  stand for the magnitudes of randomness,  $W_p$  and  $W_S$  are Wiener processes and correlated with each other by a positive coefficient  $\rho$  between 0 and 1, namely,  $dW_p dW_S = \rho dt$ . The net revenue (pollution damage included) at time  $t$  for region  $i$  is given as follows:

$$\mathcal{R}_i(t) = (A_i - S(t))E_i(t) - \frac{1}{2}E_i^2(t) + S(t)E_{i0} - D_i P(t), \tag{2.3}$$

where  $A_i$  is a positive constant,  $E_{i0}$  is the initial quota in the emission permits trading scheme,  $D_i$  is a positive constant and  $D_i P(t)$  is the pollution damage. We also assume that the salvage cost at the terminal time  $T$  is  $g_i(\bar{P}_i - P(T))$ , where  $g_i$  and  $\bar{P}_i$  are positive constants. The objective functional for region  $i$  is given by:

$$\sup_{E_i} \mathcal{E} \left[ \int_0^T e^{-rt} \mathcal{R}_i(t) dt \right] - g_i(\bar{P}_i - P(T))e^{-rT}, \tag{2.4}$$

where  $r$  is the risk-free discount rate. The above objective functional represents the optimal net revenue for the  $i$ th player from time  $t = 0$  to  $t = T$ . If two regions choose to cooperate, their joint objective functional control becomes:

$$\sup_{E_1, E_2} \mathcal{E} \left[ \int_0^T e^{-rt} (\mathcal{R}_1(t) + \mathcal{R}_2(t)) dt \right] - \sum_{i=1,2} g_i(\bar{P}_i - P(T))e^{-rT}. \tag{2.5}$$

We assume that the problem is defined on  $[P_{min}, P_{max}] \times [S_{min}, S_{max}] \times [0, T]$ , where  $P_{min}$  and  $P_{max}$  are minimum and maximum stock of pollutions,  $S_{min}$  and  $S_{max}$  are the minimum and maximum emission permits prices, and  $T$  is the terminal time when the game is mature. We also denote the pair  $(E_1, E_2)$  as  $E$  for simplicity.

### 2.1. Cooperative game

In a cooperative game, players aim to maximize their total accumulated revenue through adjusting their emissions dynamically. Therefore, the cooperative game model generates one HJB equation since only the sum of all players' revenues is of our concern. Applying dynamic programming principle to (2.5) and letting  $\tau = T - t$  for convenience, we obtain the PDE for the cooperative game:

$$\frac{\partial V}{\partial \tau} = \sup_{E_1, E_2} \{ \mathcal{L}^E V + F(P, S, \tau, E_1, E_2) \}, \tag{2.6}$$

where  $V$  is the joint value function. The expression of  $\mathcal{L}^E V$  is given as follows:

$$(E_1 + E_2 - \theta_p P) \frac{\partial V}{\partial P} + \frac{1}{2} \sigma_p^2 P^2 \frac{\partial^2 V}{\partial P^2} + \mu_S S \frac{\partial V}{\partial S} + \frac{1}{2} \sigma_S^2 S^2 \frac{\partial^2 V}{\partial S^2} + \rho \sigma_p \sigma_S P S \frac{\partial^2 V}{\partial P \partial S} - rV, \tag{2.7}$$

and

$$F(P, S, \tau, E_1, E_2) = (A_1 - S)E_1 + (A_2 - S)E_2 - \frac{E_1^2 + E_2^2}{2} + (E_{10} + E_{20})S - (D_1 + D_2)P, \tag{2.8}$$

with the initial condition:

$$V(P, S, 0) = -g_1(P - \bar{P}_1) - g_2(P - \bar{P}_2) = -(g_1 + g_2)P + g_1 \bar{P}_1 + g_2 \bar{P}_2. \tag{2.9}$$

The initial condition above corresponds to the salvage costs at the terminal time  $t = T$ .

### 2.2. Non-cooperative game

In contrary to cooperative games, each player in non-cooperative games is interested in maximizing its own revenue only, which means that the model derives a system of HJB equations, each of which corresponds to one player and is coupled with the other through the emission controls. In the non-cooperative modeling, we aim to find the solutions such that the equilibrium status is reached. Similar to the derivation of the HJB equation for cooperative games, the system for non-cooperative games is given as follows:

$$\begin{cases} \frac{\partial V_1}{\partial \tau} = \sup_{E_1} \{ \mathcal{L}^E V_1 + F_1(P, S, \tau, E_1) \} \\ \frac{\partial V_2}{\partial \tau} = \sup_{E_2} \{ \mathcal{L}^E V_2 + F_2(P, S, \tau, E_2) \}, \end{cases} \tag{2.10}$$

where  $V_i$  ( $i = 1, 2$ ) is the value function for the  $i$ th player, and the linear differential operator  $\mathcal{L}^E$  on  $V_i$ , namely,  $\mathcal{L}^E V_i$ , is given as follows:

$$(E_1 + E_2 - \theta_p P) \frac{\partial V_i}{\partial P} + \frac{1}{2} \sigma_p^2 P^2 \frac{\partial^2 V_i}{\partial P^2} + \mu_S S \frac{\partial V_i}{\partial S} + \frac{1}{2} \sigma_S^2 S^2 \frac{\partial^2 V_i}{\partial S^2} + \rho \sigma_p \sigma_S P S \frac{\partial^2 V_i}{\partial P \partial S} - rV_i \tag{2.11}$$

and

$$F_i(P, S, \tau, E_i) = (A_i - S)E_i - \frac{E_i^2}{2} + E_{i0}S - D_iP \tag{2.12}$$

with the initial conditions

$$V_i(P, S, 0) = -g_i(P - \bar{P}_i), \tag{2.13}$$

which correspond to the salvage costs at  $t = T$  for players 1 and 2, respectively.

2.3. Boundary conditions

We present the boundary conditions, which are also used in [11], for PDEs from cooperative and non-cooperative games. First, we discuss the cooperative game. The boundary conditions at  $P_{min}$  and  $P_{max}$  are determined by multiplying the discount factor  $e^{-r\tau}$  to the initial condition:  $V(P_{min}, S, \tau) = V(P_{min}, S, 0)e^{-r\tau}$  and  $V(P_{max}, S, \tau) = V(P_{max}, S, 0)e^{-r\tau}$ . For  $S = S_{min}, S_{max}$ , condition (2.2) is canceled, which gives an one dimensional HJB equation:

$$\frac{\partial V}{\partial \tau} = \sup_{E_1, E_2} \left\{ (E_1 + E_2 - \theta_p P) \frac{\partial V}{\partial P} + \frac{1}{2} \sigma_p^2 P^2 \frac{\partial^2 V}{\partial P^2} + F(P, S_m, E_1, E_2) \right\}, \tag{2.14}$$

where  $S_m \in \{S_{min}, S_{max}\}$ .

For the non-cooperative game, we have  $V_i(P_{min}, S, \tau) = V_i(P_{min}, S, 0)e^{-r\tau}$  and  $V_i(P_{max}, S, \tau) = V_i(P_{max}, S, 0)e^{-r\tau}$ , where  $e^{-r\tau}$  is the discount factor. Similarly, we can cancel condition (2.2) when  $S = S_{min}, S_{max}$  and obtain one dimensional HJB equation for each region.

3. Numerical method for the cooperative game

In this section, we propose a finite difference scheme for solving (2.6) derived from the cooperative game. We will prove that our discretization for the equation is consistent, monotone, and stable and guarantees convergence to viscosity solution. Our discretization combines the narrow stencil [22] with the wide stencil [20] to make sure the conditions (consistency, monotonicity and stability) for viscosity solution are satisfied.

We set up some notations before discussing details about discretization. We use the uniform grid  $\{P_i\}_{i=0}^{N_p} \times \{S_j\}_{j=0}^{N_s}$  for the space discretization over  $\Omega = [P_{min}, P_{max}] \times [S_{min}, S_{max}]$ , where  $P_i = P_{min} + i \cdot \Delta P$  with  $\Delta P = \frac{P_{max} - P_{min}}{N_p}$  and  $S_j = S_{min} + j \cdot \Delta S$  with  $\Delta S = \frac{S_{max} - S_{min}}{N_s}$ . The time stepping is also discretized uniformly by  $\{\tau^n\}_{n=0}^{N_\tau}$  where  $\tau^n = n \cdot \Delta \tau$  with  $\Delta \tau = \frac{T}{N_\tau}$ . Let  $V_{i,j}^n$  be the numerical solution to  $V$  at  $(P_i, S_j, \tau^n)$ . We assume that there exists a mesh scale parameter  $h$  such that  $\Delta P = C_p h$ ,  $\Delta S = C_s h$  and  $\Delta \tau = C_\tau h$ , where  $C_p, C_s$  and  $C_\tau$  are  $h$ -independent constants.

3.1. Narrow stencil discretization

In this section, we give the details about how to discretize the HJB equation using the narrow stencil, how to select the appropriate method to satisfy the condition for viscosity, and the scenario when using the narrow stencil fails to satisfy the viscosity conditions.

The discretization of the derivative with respect to time is given as follows:

$$\frac{\partial V}{\partial \tau}(P_i, S_j, \tau^n) = \frac{V_{i,j}^{n+1} - V_{i,j}^n}{\Delta \tau} + O(\Delta \tau) = \frac{V_{i,j}^{n+1} - V_{i,j}^n}{\Delta \tau} + O(h). \tag{3.1}$$

For the first order partial derivatives with respect to space variables ( $\frac{\partial V}{\partial P}$  and  $\frac{\partial V}{\partial S}$ ), one of the forward, backward, and central difference methods is selected. Note that each method may not satisfy viscosity conditions at all grid points, which means methods with lower order of accuracy, the forward and backward, may be used in favor of central differencing. As the central differencing gives the second order accuracy while the other two give first order accuracy, we approximate first order partial derivatives using central differencing as much as possible. The detailed selection criterion will be described later.

We apply central differencing to the second partial derivatives:

$$\frac{\partial^2 V_C}{\partial P^2}(P_i, S_j, \tau^n) = \frac{V_{i+1,j}^{n+1} - 2V_{i,j}^{n+1} + V_{i-1,j}^{n+1}}{(\Delta P)^2} + O(h^2), \tag{3.2}$$

$$\frac{\partial^2 V_C}{\partial S^2}(P_i, S_j, \tau^n) = \frac{V_{i,j+1}^{n+1} - 2V_{i,j}^{n+1} + V_{i,j-1}^{n+1}}{(\Delta S)^2} + O(h^2). \tag{3.3}$$

Given  $\rho \geq 0$ , the cross second partial derivative can be approximated using the seven point stencil:

$$\frac{\partial^2 V}{\partial P \partial S} = \frac{2V_{i,j}^{n+1} + V_{i+1,j+1}^{n+1} + V_{i-1,j-1}^{n+1} - V_{i+1,j}^{n+1} - V_{i-1,j}^{n+1} - V_{i,j+1}^{n+1} - V_{i,j-1}^{n+1}}{2\Delta P \Delta S} + O(h^2). \tag{3.4}$$

**Table 3.1**  
Values for  $\beta_{i,j}^{P,k}$  and  $\beta_{i,j}^{S,k}$  where  $\beta_{i,j}^P = \frac{E_i + E_j - \theta_p P_i}{\Delta P}$ ,  $\beta_{i,j}^S = \frac{\mu_S S_j}{\Delta S}$ .

Method	$\beta_{i,j}^{P,-1}$	$\beta_{i,j}^{P,0}$	$\beta_{i,j}^{P,1}$	$\beta_{i,j}^{S,-1}$	$\beta_{i,j}^{S,0}$	$\beta_{i,j}^{S,1}$
Forward	0	$\beta_{i,j}^P$	$\beta_{i,j}^P$	0	$\beta_{i,j}^S$	$\beta_{i,j}^S$
Backward	$-\beta_{i,j}^P$	$-\beta_{i,j}^P$	0	$-\beta_{i,j}^S$	$-\beta_{i,j}^S$	0
Central	$-\beta_{i,j}^P/2$	0	$\beta_{i,j}^P/2$	$-\beta_{i,j}^S/2$	0	$\beta_{i,j}^S/2$

The discretization for the case  $\rho < 0$  is done by changing the two corner grid points in the seven point stencil. In this paper, we assume that  $\rho \geq 0$  holds for the models.

We proceed to describe how to choose the differencing method for first order partial derivatives and the condition for using narrow stencil discretization. Let  $L^E$  be the discrete linear operator on  $V_{i,j}^n$ . We have

$$L^E V_{i,j}^n = (\alpha_{i,j}^P + \beta_{i,j}^{P,-1} - \gamma_{i,j})V_{i-1,j}^n + (\alpha_{i,j}^P + \beta_{i,j}^{P,1} - \gamma_{i,j})V_{i+1,j}^n + (\alpha_{i,j}^S + \beta_{i,j}^{S,-1} - \gamma_{i,j})V_{i,j-1}^n + (\alpha_{i,j}^S + \beta_{i,j}^{S,1} - \gamma_{i,j})V_{i,j+1}^n + \gamma_{i,j}(V_{i+1,j+1}^n + V_{i-1,j-1}^n) - (2\alpha_{i,j}^P + \beta_{i,j}^{P,0} + 2\alpha_{i,j}^S + \beta_{i,j}^{S,0} - 2\gamma_{i,j} + r)V_{i,j}^n, \tag{3.5}$$

where  $\alpha_{i,j}^P = \frac{(\sigma_P P_i)^2}{2(\Delta P)^2}$ ,  $\alpha_{i,j}^S = \frac{(\sigma_S S_j)^2}{2(\Delta S)^2}$  and  $\gamma_{i,j} = \frac{\rho P_i S_j \sigma_P \sigma_S}{2\Delta P \Delta S} \geq 0$ . The coefficients  $\beta_{i,j}^{P,k}$  ( $k = -1, 0, 1$ ) and  $\beta_{i,j}^{S,k}$  ( $k = -1, 0, 1$ ) are determined by the finite differencing method needed for the first order partial derivatives. Table 3.1 shows their values corresponding to the three differencing methods.

Note that the positive coefficient condition is sufficient to ensure monotonicity [23]. The positive coefficient condition [23] is defined as follows:

$$\alpha_{i,j}^P + \beta_{i,j}^{P,-1} - \gamma_{i,j} \geq 0, \quad \alpha_{i,j}^P + \beta_{i,j}^{P,1} - \gamma_{i,j} \geq 0, \tag{3.6}$$

$$\alpha_{i,j}^S + \beta_{i,j}^{S,-1} - \gamma_{i,j} \geq 0, \quad \alpha_{i,j}^S + \beta_{i,j}^{S,1} - \gamma_{i,j} \geq 0, \tag{3.7}$$

$$\gamma_{i,j} \geq 0, \quad 2\alpha_{i,j}^P + \beta_{i,j}^{P,0} + 2\alpha_{i,j}^S + \beta_{i,j}^{S,0} - 2\gamma_{i,j} + r \geq 0. \tag{3.8}$$

From Table 3.1, we observe that

$$\beta_{i,j}^{P,-1} + \beta_{i,j}^{P,1} = \beta_{i,j}^{P,0}, \quad \beta_{i,j}^{S,-1} + \beta_{i,j}^{S,1} = \beta_{i,j}^{S,0}. \tag{3.9}$$

Adding all the equations in (3.6) and (3.7) together and combining the equations in (3.9), we obtain:

$$2\alpha_{i,j}^P + \beta_{i,j}^{P,0} + 2\alpha_{i,j}^S + \beta_{i,j}^{S,0} - 2\gamma_{i,j} \geq 2\gamma_{i,j}. \tag{3.10}$$

Therefore, (3.8) holds since  $\gamma_{i,j} \geq 0$  and (3.6) and (3.7) are satisfied.

We present the algorithm for selecting differencing method for the first order partial derivatives. This algorithm guarantees that (3.6) and (3.7) are satisfied under the following conditions:

$$\alpha_{i,j}^P - \gamma_{i,j} \geq 0, \quad \alpha_{i,j}^S - \gamma_{i,j} \geq 0, \tag{3.11}$$

namely,

$$\frac{(\sigma_P P_i)^2}{2(\Delta P)^2} \geq \frac{\rho P_i S_j \sigma_P \sigma_S}{2\Delta P \Delta S} = \gamma_{i,j}, \quad \frac{(\sigma_S S_j)^2}{2(\Delta S)^2} \geq \frac{\rho P_i S_j \sigma_P \sigma_S}{2\Delta P \Delta S} = \gamma_{i,j}. \tag{3.12}$$

---

**Algorithm 1** Selecting differencing method for first order derivatives with respect to  $K = P, S$ .

---

- 1: **if**  $\alpha_{i,j}^K - \beta_{i,j}^K/2 - \gamma_{i,j} \geq 0$  and  $\alpha_{i,j}^K + \beta_{i,j}^K/2 - \gamma_{i,j} \geq 0$  **then**
  - 2:     Use central differencing:
  - 3:      $\beta_{i,j}^{K,-1} \leftarrow -\beta_{i,j}^K/2$ ,  $\beta_{i,j}^{K,0} \leftarrow 0$ ,  $\beta_{i,j}^{K,1} \leftarrow \beta_{i,j}^K/2$
  - 4: **else if**  $\beta_{i,j}^K \geq 0$  **then**
  - 5:     Use forward differencing:
  - 6:      $\beta_{i,j}^{K,-1} \leftarrow 0$ ,  $\beta_{i,j}^{K,0} \leftarrow \beta_{i,j}^K$ ,  $\beta_{i,j}^{K,1} \leftarrow \beta_{i,j}^K$
  - 7: **else**
  - 8:     Use backward differencing:
  - 9:      $\beta_{i,j}^{K,-1} \leftarrow -\beta_{i,j}^K$ ,  $\beta_{i,j}^{K,0} \leftarrow -\beta_{i,j}^K$ ,  $\beta_{i,j}^{K,1} \leftarrow 0$
- 

In the above algorithm, the central differencing is used only if conditions (3.6) and (3.7) are verified to be true. One of the forward and backward differencing methods is selected to guarantee that  $\beta_{i,j}^{K,+1/-1} \geq 0$  ( $K = P, S$ ). Since we assume that  $\alpha_{i,j}^K - \gamma_{i,j} \geq 0$  ( $K = P, S$ ), conditions (3.6) and (3.7) also hold when using the forward/backward differencing properly.

Unfortunately, condition (3.11) may not be satisfied in general due to the existence of the cross derivative. In order to address this issue, we implement a wide stencil discretization on those grid points at which condition (3.11) fails to hold.

3.2. Wide stencil discretization

In this section, we describe how to implement wide stencil discretization when condition (3.11) cannot be satisfied. As we can see, the problem resides on the cross second order partial derivative. The main idea of the wide stencil is to rotate the local coordinate system to eliminate the cross derivative.

Consider the following virtual clockwise rotation by  $\theta_{i,j}$  at grid point  $(P_i, S_j)$ :

$$\mathbf{S} = \begin{pmatrix} P \\ S \end{pmatrix} = \begin{pmatrix} \cos \theta_{i,j} & -\sin \theta_{i,j} \\ \sin \theta_{i,j} & \cos \theta_{i,j} \end{pmatrix} \begin{pmatrix} Z \\ W \end{pmatrix} = \mathbf{R}_{i,j} \begin{pmatrix} Z \\ W \end{pmatrix} = \mathbf{R}_{i,j} \mathbf{W}, \tag{3.13}$$

where  $\theta_{i,j} = \frac{1}{2} \tan^{-1} \left( \frac{2\rho\sigma_P\sigma_S P_i S_j}{(\sigma_P P_i)^2 - (\sigma_S S_j)^2} \right)$ . After the above rotation, the second order derivatives in (2.7) becomes

$$\frac{1}{2} \sigma_P^2 P_i^2 \frac{\partial^2 V}{\partial P^2} + \frac{1}{2} \sigma_S^2 S_j^2 \frac{\partial^2 V}{\partial S^2} + \rho \sigma_P \sigma_S P_i S_j \frac{\partial^2 V}{\partial P \partial S} = a_{i,j} \frac{\partial^2 V}{\partial Z^2} + b_{i,j} \frac{\partial^2 V}{\partial W^2}, \tag{3.14}$$

where  $\mathcal{V} = \mathcal{V}(Z, W, \tau) = V(P, S, \tau)$  and

$$a_{i,j} = \frac{(\sigma_P \cos(\theta_{i,j}) P_i)^2}{2} + \rho \sigma_P \sigma_S P_i S_j \sin \theta_{i,j} \cos \theta_{i,j} + \frac{(\sigma_S \sin(\theta_{i,j}) S_j)^2}{2}, \tag{3.15}$$

$$b_{i,j} = \frac{(\sigma_P \sin(\theta_{i,j}) P_i)^2}{2} - \rho \sigma_P \sigma_S P_i S_j \sin \theta_{i,j} \cos \theta_{i,j} + \frac{(\sigma_S \cos(\theta_{i,j}) S_j)^2}{2}. \tag{3.16}$$

The coefficients of the second order derivatives with respect to  $Z$  and  $W$  are larger as the correlation factor  $\rho$  is assumed to be non-negative and less than 1. We note that  $a_{i,j}, b_{i,j}, \theta_{i,j}$  do not depend on the emission controls, which means that they can be computed in advance and re-used for each iteration and time step.

To approximate the second derivatives with respect to the virtual directions  $Z$  and  $W$ , namely,  $\frac{\partial^2 \mathcal{V}}{\partial Z^2}$  and  $\frac{\partial^2 \mathcal{V}}{\partial W^2}$ , we apply the standard three point differencing to the virtual grid along the  $Z$  and  $W$  axes. We define the following notations before explaining further:

$$\mathcal{V}^n(\mathbf{S}) = V((P, S, \tau^n), \quad \mathcal{V}^n(\mathbf{W}) = \mathcal{V}((Z, W, \tau^n)). \tag{3.17}$$

The virtual neighbor grid points of  $\mathbf{S}_{i,j} = \mathbf{R}_{i,j} \mathbf{W}_{i,j}$  are chosen as  $\mathbf{W}_{i,j} \pm \sqrt{h} \mathbf{e}_Z$  and  $\mathbf{W}_{i,j} \pm \sqrt{h} \mathbf{e}_W$  under the virtual coordinate system, where

$$\mathbf{e}_Z = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{e}_W = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{3.18}$$

Therefore, the second order partial derivatives on the right hand side of (3.14) can be approximated as:

$$a_{i,j} \frac{\mathcal{V}^n(\mathbf{W}_{i,j} + \sqrt{h} \mathbf{e}_Z) - 2\mathcal{V}^n(\mathbf{W}_{i,j}) + \mathcal{V}^n(\mathbf{W}_{i,j} - \sqrt{h} \mathbf{e}_Z)}{h} + b_{i,j} \frac{\mathcal{V}^n(\mathbf{W}_{i,j} + \sqrt{h} \mathbf{e}_W) - 2\mathcal{V}^n(\mathbf{W}_{i,j}) + \mathcal{V}^n(\mathbf{W}_{i,j} - \sqrt{h} \mathbf{e}_W)}{h} + O(h). \tag{3.19}$$

Here we choose  $\sqrt{h}$  instead of  $h$  as the size of the wide stencil to preserve consistency. Applying the transformation in (3.13)–(3.19), we express the virtual neighbor grid points under the local coordination system:

$$\mathcal{V}^n(\mathbf{W}_{i,j} \pm \sqrt{h} \mathbf{e}_Z) = V^n(\mathbf{S}_{i,j} \pm \sqrt{h}(\mathbf{R}_{i,j})_1), \tag{3.20}$$

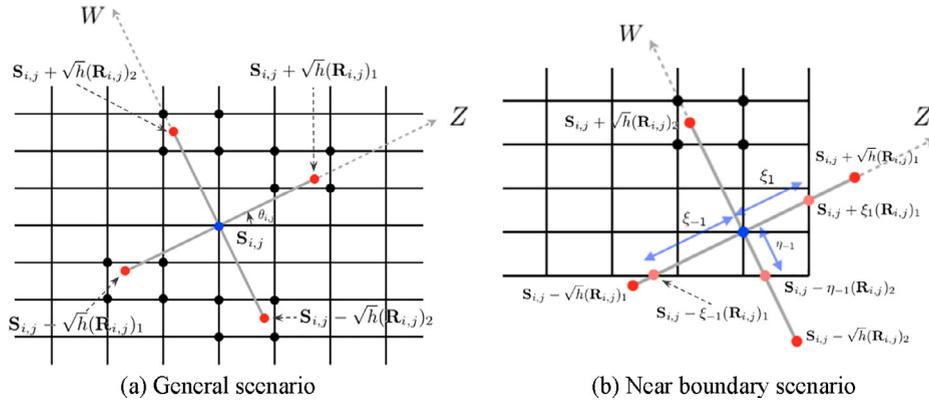
$$\mathcal{V}^n(\mathbf{W}_{i,j} \pm \sqrt{h} \mathbf{e}_W) = V^n(\mathbf{S}_{i,j} \pm \sqrt{h}(\mathbf{R}_{i,j})_2), \tag{3.21}$$

where  $(\mathbf{R}_{i,j})_k$  is the  $k$ th column of the matrix  $\mathbf{R}_{i,j}$ . Fig. 3.1(a) illustrates how virtual rotation works in general.

As we know, the virtual neighbor grid points (red ones) usually may not be grid points in the uniform grid  $\{P_i\}_{i=0}^{N_P} \times \{S_j\}_{j=0}^{N_S}$ . Instead, they may be located inside the squares in the uniform grid. We choose to approximate each virtual neighbor point using bilinear interpolation on the four grid points (black ones) of the square in which the point is located. We use  $\mathcal{J}_h V$  to denote the approximation of the value function  $V$  at those points. Therefore, we obtain

$$a_{i,j} \frac{\partial^2 \mathcal{V}}{\partial Z^2} + b_{i,j} \frac{\partial^2 \mathcal{V}}{\partial W^2} = a_{i,j} \frac{\mathcal{J}_h V^n(\mathbf{S}_{i,j} + \sqrt{h}(\mathbf{R}_{i,j})_1) - 2V^n(\mathbf{S}_{i,j}) + \mathcal{J}_h V^n(\mathbf{S}_{i,j} - \sqrt{h}(\mathbf{R}_{i,j})_1)}{h} + b_{i,j} \frac{\mathcal{J}_h V^n(\mathbf{S}_{i,j} + \sqrt{h}(\mathbf{R}_{i,j})_2) - 2V^n(\mathbf{S}_{i,j}) + \mathcal{J}_h V^n(\mathbf{S}_{i,j} - \sqrt{h}(\mathbf{R}_{i,j})_2)}{h} + O(h). \tag{3.22}$$

Here we can easily see that the order of the error becomes  $O(1)$  if we use  $h$  as the wide stencil size. Since the bilinear interpolation error is of  $O(h^2)$ , which appears in the numerator in (3.22), the denominator has to be of  $O(h)$  in order to limit



**Fig. 3.1.** Virtual rotation of the local coordinate system at  $(P_i, S_j)$ . (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

the error to  $O(h)$ . Assume that the virtual point  $(P, S)$  falls in the  $(p, s)$ th grid, which is formed by the grid points  $(P_p, S_s)$ ,  $(P_{p+1}, S_s)$ ,  $(P_{p+1}, S_{s+1})$  and  $(P_p, S_{s+1})$ . The value function at this virtual point can be interpolated as:

$$\sum_{i=0,1, j=0,1} w_{p+i, s+j} V_{p+i, s+j}^n, \tag{3.23}$$

where

$$w_{p+i, s+j} = \frac{(-1)^{i+j} (P_{p+1-i} - P)(S_{q+1-j} - S)}{\Delta P \Delta S}. \tag{3.24}$$

Now we consider the case that the grid point  $(P_i, S_j)$  is so close to the boundary  $\partial\Omega$  that some of its virtual neighbor grid points fall outside the domain, which is illustrated in Fig. 3.1(b). In this case, those virtual points cannot be approximated by interpolating their surrounding grid points. We resolve the issue by shrinking the virtual point to the boundary along the corresponding axis. For simplicity, we illustrate how to approximate  $\frac{\partial^2 \gamma}{\partial Z^2}$ . The approximation of  $\frac{\partial^2 \gamma}{\partial W^2}$  is similar. Two situations can appear when the grid point is close to the boundary: only one of the virtual neighbor grid points goes beyond the boundary; both of them go beyond the boundary. We will discuss how to handle these two situations.

Assume only  $S_{i,j} + \sqrt{h}(\mathbf{R}_{i,j})_1$  falls outside the domain. We shrink it along  $Z$  axis so that  $S_{i,j} + \eta(\mathbf{R}_{i,j})_1$  is on the boundary where  $\eta > 0$ . Note that  $S_{i,j} + \eta(\mathbf{R}_{i,j})_1$  may not coincide with any grid point on the boundary and the value function at other non-grid boundary points may be unknown. Therefore, we may have to approximate it by interpolating its two neighbor grid points. In either case, we use  $\mathcal{J}_h V(S_{i,j} + \eta(\mathbf{R}_{i,j})_1)$  to denote the exact or interpolated value of the value function at this point.

Applying the standard three point stencil, the term  $\frac{\partial^2 \gamma}{\partial Z^2}$  can be approximated as:

$$\frac{\partial^2 \gamma}{\partial Z^2}(Z_i, W_j) \approx \frac{\frac{\mathcal{J}_h V^n(S_{i,j} + \eta(\mathbf{R}_{i,j})_1) - V^n(S_{i,j})}{\eta} - \frac{V^n(S_{i,j}) - \mathcal{J}_h V^n(S_{i,j} - \sqrt{h}(\mathbf{R}_{i,j})_1)}{\sqrt{h}}}{\frac{\eta + \sqrt{h}}{2}}. \tag{3.25}$$

Now we consider the second scenario. Assume that the two shrunk points are  $S_{i,j} + \eta_1(\mathbf{R}_{i,j})_1$  and  $S_{i,j} - \eta_{-1}(\mathbf{R}_{i,j})_1$  respectively, where  $\eta_k > 0$  ( $k = \pm 1$ ). In this case, we have

$$\frac{\partial^2 \gamma}{\partial Z^2}(Z_i, W_j) \approx \frac{\frac{\mathcal{J}_h V^n(S_{i,j} + \eta_1(\mathbf{R}_{i,j})_1) - V^n(S_{i,j})}{\eta_1} - \frac{V^n(S_{i,j}) - \mathcal{J}_h V^n(S_{i,j} - \eta_{-1}(\mathbf{R}_{i,j})_1)}{\eta_{-1}}}{\frac{\eta_1 + \eta_{-1}}{2}}. \tag{3.26}$$

To summarize, we give the following algorithm to illustrate how to approximate the second order partial derivatives terms ( $\frac{\partial^2 \gamma}{\partial Z^2}$  and  $\frac{\partial^2 \gamma}{\partial W^2}$ ) in the wide stencil.

It can be easily verified from the above algorithm that the coefficient of  $V_{i,j}^n$  is always non-positive while the coefficients of other  $V_{i_1, j_1}^n$  ( $i_1 \neq i, j_1 \neq j$ ) are non-negative as  $a_{i,j} \geq 0, b_{i,j} \geq 0$  and the weights (3.24) for interpolation are non-negative.

We now deal with the first order partial derivatives in (2.7) when using the wide stencil. As there is no guarantee that the negative coefficient of the value function at each neighbor grid point can be offset by the corresponding positive coefficient from the second order partial derivative discretization, the central differencing cannot be used. We choose either the forward or the backward differencing to make sure that the coefficients of  $V_{i_1, j_1}^n$  ( $i_1 \neq i, j_1 \neq j$ ) are non-negative while the coefficient of  $V_{i,j}^n$  is non-positive. In summary, we are able to obtain the non-negative coefficients for  $V_{i_1, j_1}^n$  ( $i_1 \neq i, j_1 \neq j$ ) and the non-positive coefficient for  $V_{i,j}^n$ . As  $\sqrt{h} > h$  when  $h < 1$ , we let  $h$  be small enough so that the virtual neighbor grid points

**Algorithm 2** Approximation of  $\frac{\partial^2 v}{\partial K^2}$  in the wide stencil,  $(k, K) = (1, Z), (2, W)$ .

Let  $\eta_1 = \sqrt{h}$ ,  $\eta_{-1} = \sqrt{h}$ .

**if**  $\mathbf{S}_{i,j} + \eta_1(\mathbf{R}_{i,j})_k$  falls outside the domain  $\Omega$  **then**

Shrink along the corresponding axis and direction, and find  $\eta_1 > 0$  such that:

$\mathbf{S}_{i,j} + \eta_1(\mathbf{R}_{i,j})_k$  is on the boundary  $\partial\Omega$ .

**if**  $\mathbf{S}_{i,j} - \eta_{-1}(\mathbf{R}_{i,j})_k$  falls outside the domain  $\Omega$  **then**

Shrink along the corresponding axis and direction, and find  $\eta_{-1} > 0$  such that:

$\mathbf{S}_{i,j} - \eta_{-1}(\mathbf{R}_{i,j})_k$  is on the boundary  $\partial\Omega$ .

Approximate the second order partial derivative in the wide stencil by:

$$\frac{\frac{\mathcal{J}_h V^n(\mathbf{S}_{i,j} + \eta_1(\mathbf{R}_{i,j})_k) - V^n(\mathbf{S}_{i,j})}{\eta_1} - \frac{V^n(\mathbf{S}_{i,j}) - \mathcal{J}_h V^n(\mathbf{S}_{i,j} - \eta_{-1}(\mathbf{R}_{i,j})_k)}{\eta_{-1}}}{\frac{\eta_1 + \eta_{-1}}{2}}. \tag{3.27}$$

$\mathbf{S}_{i,j} \pm \sqrt{h}(\mathbf{R}_{i,j})_1$  and  $\mathbf{S}_{i,j} \pm \sqrt{h}(\mathbf{R}_{i,j})_2$  do not fall inside the grid that contains  $(P_i, S_j)$ . Therefore, the coefficients for the value function at the grid points used for interpolation in (3.24) are non-negative. The expression of  $L^E V^n$  is given as follows:

$$\begin{aligned} L^E V^n &= \frac{2a_{i,j}}{(\xi_1 + \xi_{-1})\xi_1} \mathcal{J}_h V^n(\mathbf{S}_{i,j} + \xi_1(\mathbf{R}_{i,j})_1) + \frac{2a_{i,j}}{(\xi_1 + \xi_{-1})\xi_{-1}} \mathcal{J}_h V^n(\mathbf{S}_{i,j} - \xi_{-1}(\mathbf{R}_{i,j})_1) \\ &+ \frac{2b_{i,j}}{(\eta_1 + \eta_{-1})\eta_1} \mathcal{J}_h V^n(\mathbf{S}_{i,j} + \eta_1(\mathbf{R}_{i,j})_1) + \frac{2b_{i,j}}{(\eta_1 + \eta_{-1})\eta_{-1}} \mathcal{J}_h V^n(\mathbf{S}_{i,j} - \eta_{-1}(\mathbf{R}_{i,j})_2) \\ &+ 1_{\beta_{i,j}^p \geq 0} \beta_{i,j}^p V_{i+1,j}^n - 1_{\beta_{i,j}^p < 0} \beta_{i,j}^p V_{i-1,j}^n + 1_{\beta_{i,j}^s \geq 0} \beta_{i,j}^s V_{i,j+1}^n - 1_{\beta_{i,j}^s < 0} \beta_{i,j}^s V_{i,j-1}^n \\ &- \left( 1_{\beta_{i,j}^p \geq 0} \beta_{i,j}^p - 1_{\beta_{i,j}^p < 0} \beta_{i,j}^p + 1_{\beta_{i,j}^s \geq 0} \beta_{i,j}^s - 1_{\beta_{i,j}^s < 0} \beta_{i,j}^s + \frac{4a_{i,j}}{(\xi_1 + \xi_{-1})\xi_1} + \frac{4b_{i,j}}{(\eta_1 + \eta_{-1})\eta_1} + r \right) V_{i,j}^n, \end{aligned} \tag{3.28}$$

where  $\xi$  and  $\eta$  represents the  $\eta$  in Algorithm 2 for directions  $Z$  and  $W$ , respectively.

### 3.3. The discretized equations in matrix form

We express the discretization methods described in the previous sections in matrix form as it is convenient for us to describe policy iteration and prove the convergence to the viscosity solution. First, we write the unknowns in vector form:

$$\mathbf{V}^n = (V_{1,1}^n, V_{2,1}^n, \dots, V_{N_p-1,1}^n, \dots, V_{1,N_s-1}^n, \dots, V_{N_p-1,N_s-1}^n)^T,$$

where the  $l$ th entry of  $\mathbf{V}^n$  is

$$\mathbf{V}_l^n = V_{i,j}^n, \quad l = i - 1 + (j - 1)(N_p - 1). \tag{3.29}$$

The size of the vector of unknowns is  $N = (N_p - 1)(N_s - 1)$ . We write the discretized equation for each grid point  $(P_i, S_j)$  as  $L^E V^n = \mathbf{L}_l^E \mathbf{V}^n + \mathbf{B}_l^E$  where  $\mathbf{L}_l^E$  is a row vector,  $\mathbf{B}_l^E$  is a column vector used to contain all boundary values involved and  $l$  is defined in (3.29). Note that  $\mathbf{B}_l^E$  is determined by the control set  $E$  as the differencing method for one of the first order partial derivative is control dependent. We define the  $N \times N$  matrix  $\mathbf{L}^E$  where its  $l$ th row is  $\mathbf{L}_l^E$ . Let  $\mathbf{B}^E$  be an  $N \times 1$  column vector with  $\mathbf{B}_l^E$  as its  $l$ th row. We have the following coupled linear equation:

$$\frac{\mathbf{V}^{n+1} - \mathbf{V}^n}{\Delta \tau} = \sup_E \{ \mathbf{L}^E \mathbf{V}^{n+1} + \mathbf{B}^E + \mathbf{F}^E \}, \tag{3.30}$$

where the column vector  $\mathbf{F}^E$  is formed by  $F(P, S, E)$  at each inner grid point. Eq. (3.30) can also be written as:

$$(\mathbf{I} - \Delta \tau \mathbf{L}^{E^*}) \mathbf{V}^{n+1} = \Delta \tau (\mathbf{B}^{E^*} + \mathbf{F}^{E^*}) + \mathbf{V}^n, \tag{3.31}$$

where

$$E^* = \operatorname{argsup}_E \{ \mathbf{L}^E \mathbf{V}^{n+1} + \mathbf{B}^E + \mathbf{F}^E \}. \tag{3.32}$$

Eq. (3.31) represents the  $n$ th time step discretization, where  $\mathbf{V}^n$  is known and the objective is to solve for  $\mathbf{V}^{n+1}$  and  $E^*$ .

### 3.4. Policy iteration

By (3.31), the unknowns  $\mathbf{V}^{n+1}$  at the  $n$ th time step and the control  $E^*$  are dependent on each other. The standard methods for solving linear systems cannot be applied in this case. We introduce a method called policy iteration to solve this coupled system.

The main idea of policy iteration is to fix either of the two unknowns in the coupled equation in order to solve the other unknown, and then alternate. Specifically, we fix the value function by giving it an initial guess. We then compute the optimal control, which can be fixed and used to compute the new value function. We will discuss how to solve controls in details later. With the newly obtained control, the coupled equation becomes an ordinary uncoupled linear system, which can be solved using numerical methods for linear systems. Then, we obtain the new value for the value function. This process is repeated until the difference of the values for the unknowns between two iteration steps is small enough.

The algorithm description is given in Algorithm 3. The convergence of policy iteration has been proved in [23].

---

**Algorithm 3** Policy iteration for solving the equation at the  $n$ th time step.

---

- 1: Let  $\hat{\mathbf{V}}^k$  be the numerical approximation to  $\mathbf{V}^{n+1}$  after the  $k$ th iteration.
- 2: Give an initial guess to  $\hat{\mathbf{V}}^0$ .
- 3: **for**  $k = 0, 1, 2, \dots$  until convergence ( $\|\hat{\mathbf{V}}^{k+1} - \hat{\mathbf{V}}^k\|$  is small enough) **do**
- 4: Find the optimal control  $E^k$  such that

$$E^k = \operatorname{argsup}_E \{ \mathbf{L}^E \hat{\mathbf{V}}^k + \mathbf{B}^E + \mathbf{F}^E \}. \tag{3.33}$$

- 5: Solve the linear system

$$(\mathbf{I} - \Delta \tau \mathbf{L}^{E^k}) \hat{\mathbf{V}}^{k+1} = \Delta \tau (\mathbf{B}^{E^k} + \mathbf{F}^{E^k}) + \mathbf{V}^n. \tag{3.34}$$

- 6: The numerical solutions to  $\mathbf{V}^{n+1}$  in (3.31) and  $E^*$  in (3.32) are given by  $\hat{\mathbf{V}}^{k+1}$  and  $E^k$ , respectively.
- 

Now we describe how to find the optimal control in (3.33) given the value function  $\hat{\mathbf{V}}^k$ . We note that (3.11) determines whether the wide stencil will be used or not. We discuss how to find the control for each row based on which of the two stencils is used.

First we consider the narrow stencil case. The condition for selecting the stencil (3.11) does not rely on the controls. We focus on how to obtain the control by considering the three differencing methods. The condition for using each differencing method is indicated in Algorithm 1 (line 1, line 4 and line 7). Solving those inequalities with respect to the controls  $E_1$  and  $E_2$  generates inequalities with respect to  $E_1 + E_2$  of the following form:

$$\alpha \leq E_1 + E_2 \leq \beta, \text{ or } E_1 + E_2 \leq \beta, \text{ or } \alpha \leq E_1 + E_2 \tag{3.35}$$

The space for controls  $(E_1, E_2)$  can be divided into three regions, each of which is defined by one of the inequalities in (3.35) and supports the corresponding differencing method. We find the maximum value for the terms in the parenthesis on the right hand side of (3.33) for each region. Then, the control  $E^k$  is the control for the region which has the maximum value among the three. Finding the control in a region is not difficult as the terms to be optimized are of quadratic form with respect to  $E_1$  and  $E_2$ :

$$AE_1 - \frac{1}{2}E_1^2 + BE_2 - \frac{1}{2}E_2^2. \tag{3.36}$$

The optimal control is either  $(\frac{A}{2}, \frac{B}{2})$  or some point on the boundary.

As for the wide stencil, the situation is simpler since ruling out of the central differencing leaves only the forward and the backward differencing methods to be considered. The space for the controls  $(E_1, E_2)$  is divided into two regions instead of three. The term to be optimized still has quadratic form as in (3.36). The solution is similar: either  $(\frac{A}{2}, \frac{B}{2})$  or some point on the boundary of the two regions.

### 3.5. Convergence analysis

In this section, we will prove that our method guarantees convergence to the viscosity solution by proving the scheme to be consistent, stable and monotone.

*Consistency.* Before giving the definition of consistency in the viscosity sense, we define some notations for convenience. Moving the right hand side of (2.6) to the left hand side, we denote the resulting equation as:

$$\mathcal{F}V \equiv \mathcal{F}(\mathbf{p}, V(\mathbf{p}), DV(\mathbf{p}), D^2V(\mathbf{p})) = \frac{\partial V}{\partial \tau}(\mathbf{p}) - \sup_{E_1, E_2} \{ \mathcal{L}^E V(\mathbf{p}) + F(\mathbf{p}, E_1, E_2) \} = 0, \tag{3.37}$$

where  $\mathbf{p} = (P, S, \tau)$ ,  $DV$  is the gradient vector of  $V$  with respect to  $(P, S)$ ,  $D^2V$  is Hessian matrix of  $V$  with respect to  $(P, S)$  and  $\mathcal{F}$  is an operator applied to the value function  $V$ . It is evident that  $\mathcal{F}V = 0$  holds when  $V$  satisfies (2.6).

Before proceeding to the definition of consistency, we recall the limit superior and limit inferior of a real-valued function. We denote the limit superior and inferior of real-valued function  $\mathcal{F}V$  with respect to  $p$  as  $\mathcal{F}^*V \equiv \mathcal{F}^*(\mathbf{p}, V(\mathbf{p}), DV(\mathbf{p}), D^2V(\mathbf{p}))$  and  $\mathcal{F}_*V \equiv \mathcal{F}_*(\mathbf{p}, V(\mathbf{p}), DV(\mathbf{p}), D^2V(\mathbf{p}))$ , respectively. Their definitions are given as follows:

$$\mathcal{F}^*(\mathbf{p}, V(\mathbf{p}), DV(\mathbf{p}), D^2V(\mathbf{p})) = \limsup_{\mathbf{q} \rightarrow \mathbf{p}} (\mathcal{F}(\mathbf{q}, V(\mathbf{q}), DV(\mathbf{q}), D^2V(\mathbf{q}))), \tag{3.38}$$

$$\mathcal{F}_*(\mathbf{p}, V(\mathbf{p}), DV(\mathbf{p}), D^2V(\mathbf{p})) = \liminf_{\mathbf{q} \rightarrow \mathbf{p}} (\mathcal{F}(\mathbf{q}, V(\mathbf{q}), DV(\mathbf{q}), D^2V(\mathbf{q}))). \tag{3.39}$$

Now we give the definition of consistency in the viscosity sense.

**Definition 3.1.** (Consistency) Let  $\mathcal{F}^h$  be the numerical scheme for the PDE  $\mathcal{F}V = 0$ . Namely, we denote  $\mathcal{F}^h(\mathbf{p}_{i,j}^{n+1}, V_{i,j}^{n+1}, \{V_{p,q}\}_{(p,q) \neq (i,j)}, V_{i,j}^n) = 0$  as the discretized equation at grid point  $(P_i, S_j)$  for the  $(n + 1)$ th time step with mesh parameter of  $h$ . The numerical scheme  $\mathcal{F}^h$  is consistent in the viscosity sense if, for any  $C^\infty$  real-valued function  $\phi$  and any point  $\hat{\mathbf{p}}$  in  $\Omega$ , the following conditions hold:

$$\limsup_{\substack{h \rightarrow 0 \\ \epsilon \rightarrow 0 \\ \mathbf{p}_{i,j}^{n+1} \rightarrow \hat{\mathbf{p}}}} \mathcal{F}^h(\mathbf{p}_{i,j}^{n+1}, \phi_{i,j}^{n+1} + \epsilon, \{\phi_{p,q}^{n+1} + \epsilon\}_{(p,q) \neq (i,j)}, \phi_{i,j}^n + \epsilon) \leq \mathcal{F}^*(\hat{\mathbf{p}}, \phi(\hat{\mathbf{p}}), D\phi(\hat{\mathbf{p}}), D^2\phi(\hat{\mathbf{p}})), \tag{3.40}$$

and

$$\liminf_{\substack{h \rightarrow 0 \\ \epsilon \rightarrow 0 \\ \mathbf{p}_{i,j}^{n+1} \rightarrow \hat{\mathbf{p}}}} \mathcal{F}^h(\mathbf{p}_{i,j}^{n+1}, \phi_{i,j}^{n+1} + \epsilon, \{\phi_{p,q}^{n+1} + \epsilon\}_{(p,q) \geq (i,j)}, \phi_{i,j}^n + \epsilon) \geq \mathcal{F}_*(\hat{\mathbf{p}}, \phi(\hat{\mathbf{p}}), D\phi(\hat{\mathbf{p}}), D^2\phi(\hat{\mathbf{p}})). \tag{3.41}$$

where  $\phi_{i,j}^n = \phi(P_i, S_j, \tau^n)$ .

We prove that our numerical scheme is consistent according to the above definition by showing its local consistency.

**Lemma 3.2.** (Local consistency) Let  $\mathcal{F}^h$  denote the numerical scheme defined in (3.5) and (3.28). For any  $C^\infty$  real-valued function  $\phi$  defined on  $\Omega \times [0, T]$ , the following statement holds:

$$\begin{aligned} & \mathcal{F}^h(\mathbf{p}_{i,j}^{n+1}, \phi_{i,j}^{n+1} + \epsilon, \{\phi_{p,q}^{n+1} + \epsilon\}_{(p,q) \neq (i,j)}, \phi_{i,j}^n + \epsilon) - \mathcal{F}(\mathbf{p}_{i,j}^{n+1}, \phi(\mathbf{p}_{i,j}^{n+1}), D\phi(\mathbf{p}_{i,j}^{n+1}), D^2\phi(\mathbf{p}_{i,j}^{n+1})) \\ &= \begin{cases} O(\sqrt{h}) + O(\epsilon), & \text{the wide stencil with stencil points outside } \Omega \\ O(h) + O(\epsilon), & \text{otherwise.} \end{cases} \end{aligned} \tag{3.42}$$

**Proof.** We complete the proof by dividing the problem into several cases.

For the case that the narrow stencil is used, we apply (3.5) and note that  $\epsilon$  will be canceled by the discretization of the derivative terms. For the first order derivative terms, the relative errors after discretization could be  $O(h)$  or  $O(h^2)$ . All of the discretized second order derivative terms have the truncation error of  $O(h^2)$ . Therefore, the worst case for the order of the truncation error is  $O(h)$ . We have

$$\begin{aligned} & \mathcal{F}^h(\mathbf{p}_{i,j}^{n+1}, \phi_{i,j}^{n+1} + \epsilon, \{\phi_{p,q}^{n+1} + \epsilon\}_{(p,q) \neq (i,j)}, \phi_{i,j}^n + \epsilon) \\ &= \frac{\phi_{i,j}^{n+1} - \phi_{i,j}^n}{\Delta \tau} - \sup_E (L^E \phi^{n+1} + F(\mathbf{p}_{i,j}^{n+1}, E)) \\ &= \left( \frac{\partial \phi}{\partial \tau} \right)_{i,j}^{n+1} + O(h) - \sup_E ((\mathcal{L}^E \phi)(\mathbf{p}_{i,j}^{n+1}) + F(\mathbf{p}_{i,j}^{n+1}, E) + O(h) + O(\epsilon)) \\ &= \mathcal{F}(\mathbf{p}_{i,j}^{n+1}, \phi(\mathbf{p}_{i,j}^{n+1}), D\phi(\mathbf{p}_{i,j}^{n+1}), D^2\phi(\mathbf{p}_{i,j}^{n+1})) + O(h) + O(\epsilon). \end{aligned} \tag{3.43}$$

Now we proceed to the case that the wide stencil has to be applied. The discretization of first order derivative terms still gives at most the  $O(h)$  truncation error. We focus on the rotated second order derivative terms. According to (3.14) and (3.22), we can easily prove that the truncation error from second order terms is  $O(h)$  since each  $\epsilon$  after  $\phi$  is canceled in the discretization. Note that the term  $-r(\phi_{i,j}^{n+1} + \epsilon)$  in the discretized equation still gives the error of  $O(\epsilon)$ . Therefore, the left hand side of (3.42) becomes  $O(h) + O(\epsilon)$  in this case.

Finally, we consider the more complicated case. Assume that the wide stencil is applied but some stencil points are outside  $\Omega$ . For simplicity, we consider one of the two virtual directions: the direction along  $Z$  axis. We need to prove, replacing  $\mathcal{V}$  and  $V$  with  $\phi$  in Algorithm 2, that

$$\frac{\partial^2 \phi}{\partial Z^2}(Z_i, W_j) - \frac{\mathcal{J}_h \phi^n(\mathbf{S}_{i,j} + \eta_1(\mathbf{R}_{i,j})_k) - \phi^n(\mathbf{S}_{i,j})}{\eta_1} - \frac{\phi^n(\mathbf{S}_{i,j}) - \mathcal{J}_h \phi^n(\mathbf{S}_{i,j} - \eta_{-1}(\mathbf{R}_{i,j})_k)}{\eta_{-1}} = O(\sqrt{h}). \tag{3.44}$$

There are two cases to be considered: only one wide stencil point along this direction is outside  $\Omega$ ; both points are outside  $\Omega$ . Consider the first case and assume that the wide stencil point  $\mathbf{S}_{i,j} - \eta_{-1}(\mathbf{R}_{i,j})_k$  is beyond the boundary while the other one is inside. We have  $\eta_1 = \sqrt{h}$  while  $\eta_{-1} < \sqrt{h}$  and  $O(\eta_{-1}) = O(\sqrt{h})$ . Applying Taylor expansion, the left hand side of (3.44) becomes:

$$O(\sqrt{h} - \eta_{-1}) + O\left(\frac{h\sqrt{h} + \eta_{-1}^3}{\sqrt{h} + \eta_{-1}}\right) = O(\sqrt{h}) + O(h) = O(\sqrt{h}). \tag{3.45}$$

Now, we assume that both wide stencil points are outside  $\Omega$ . The left hand side of (3.44) becomes

$$O(\eta_1 - \eta_{-1}) + O\left(\frac{\eta_1^3 + \eta_{-1}^3}{\eta_1 + \eta_{-1}}\right) = O(\sqrt{h}) + O(h) = O(\sqrt{h}). \tag{3.46}$$

We now conclude the proof by showing that the local consistency holds for both situations in (3.42).  $\square$

**Theorem 3.3.** *The numerical schemes defined in (3.5) and (3.28) are consistent according to Definition 3.1 if all conditions in Lemma 3.2 hold.*

**Proof.** The proof is straightforward according to the definitions of limit superior and inferior.

**Stability.** The stability of a numerical scheme ensures the unconditional convergence regardless of the time step  $\Delta\tau$ . We show that the matrix  $\mathbf{I} - \Delta\tau\mathbf{L}^{E^*}$  in equation (3.31) is an M-matrix therefore its infinity norm is less than 1, which implies stability. We note that an  $n$  by  $n$  matrix  $\mathbf{A}$  is an M-matrix if the following conditions are satisfied:

- All of its diagonal entries are larger than 0:  $\mathbf{A}_{i,i} > 0$  ( $i = 1, \dots, n$ );
- All of its non-diagonal entries are non-positive:  $\mathbf{A}_{i,j} \leq 0$  ( $i \neq j$ );
- Strictly diagonally dominant:  $|\mathbf{A}_{i,i}| > \sum_{j \neq i} |\mathbf{A}_{i,j}|$  (for all  $i$ ).

The last condition is equivalent to  $\sum_j \mathbf{A}_{i,j} > 0$  (for all  $i$ ), namely the sum of all entries in the  $i$ -th row, given the previous two conditions are satisfied. We prove that the matrix  $\mathbf{I} - \Delta\tau\mathbf{L}^{E^*}$ , whenever the narrow or wide stencil is applied, is an M-matrix by showing that it satisfies the above three conditions.  $\square$

**Lemma 3.4.** *(M-matrix) The matrix  $\mathbf{I} - \Delta\tau\mathbf{L}^{E^*}$  in (3.31) is an M-Matrix.*

**Proof.** Under the narrow stencil, we can easily determine that the first two conditions are satisfied by verifying the Eqs. (3.5)–(3.8). As for the third condition, we have

$$\sum_k (\mathbf{I} - \Delta\tau\mathbf{L}^{E^*})_{l,k} = 1 + r\Delta\tau > 0, \tag{3.47}$$

where  $l$  is defined in (3.29).

Now we consider the wide stencil case. From (3.28) we know that some off-diagonal entries ( $i \pm 1, j \pm 1$ ) contribute non-positive values (see the third line of the equation). The diagonal entry ( $i, j$ ) is larger than 0 as  $a_{i,j}, b_{i,j}, \xi, \eta$  and  $r$  are positive. The value of a wide stencil point  $\mathcal{J}_h V^n(\mathbf{S}_{i,j} \pm \eta(\mathbf{R}_{i,j}))_k$  is interpolated from four grid points. We have stated that  $h$  is selected to be small enough so that only off-diagonal grid points will be used for interpolation. As the coefficients of value of the stencil points and the interpolation weights are positive, the coefficients of those interpolation values are also non-negative, which concludes the verification of the first two conditions for the wide stencil case. Now we proceed to the third condition. Assume that all the wide stencil points are inside  $\Omega$ . Note that the weights for the four interpolation grid points sum up to 1 in (3.24). Therefore, the contribution from the value function at each wide stencil point  $\mathcal{J}_h V^n(\mathbf{S}_{i,j} \pm \eta(\mathbf{R}_{i,j}))_k$  is the negation of their coefficients  $-\frac{2a_{i,j}}{(\xi_1 + \xi_{-1})\xi_1}$  or  $-\frac{2b_{i,j}}{(\eta_1 + \eta_{-1})\eta_1}$  in (3.28). Hence, the sum of all the  $l$ th entries in the matrix is:

$$\sum_k (\mathbf{I} - \Delta\tau\mathbf{L}^{E^*})_{l,k} = 1 + r\Delta\tau > 0. \tag{3.48}$$

Now we deal with the wide stencil points falling beyond  $\Omega$ . The contribution from that wide stencil point becomes zero. The sum of the  $l$ th entry becomes

$$\sum_k (\mathbf{I} - \Delta\tau\mathbf{L}^{E^*})_{l,k} = 1 + r\Delta\tau + \mathcal{Z} \frac{2a_{i,j}}{(\xi_1 + \xi_{-1})\xi_1} + \mathcal{W} \frac{2b_{i,j}}{(\eta_1 + \eta_{-1})\eta_1} > 0, \tag{3.49}$$

where the values  $\mathcal{Z}, \mathcal{W} = 0, 1, 2$ . Value 0 means both points along that direction fall beyond the boundary, 1 means only one is outside  $\Omega$  and 2 means both are inside  $\Omega$ . The proof is complete.  $\square$

**Theorem 3.5.** *The numerical schemes defined in (3.5) and (3.28) are unconditionally  $l^\infty$  stable as  $h \rightarrow 0$ , i.e.*

$$\|\mathbf{V}^n\|_\infty \leq \max(\|\mathbf{V}^0\|_\infty, \max_n \|\mathbf{B}^n + \mathbf{F}^n\|_\infty), \tag{3.50}$$

where  $\max_n \|\mathbf{F}^n\|_\infty$  and  $\max_n \|\mathbf{B}^n\|_\infty$  are bounded.

**Proof.** According to the proof of Lemma 3.4, we have

$$\|(\mathbf{I} - \Delta\tau\mathbf{L}^{E^*})^{-1}\|_\infty \leq \frac{1}{1 + r\Delta\tau} < 1. \tag{3.51}$$

Eq. (3.50) follows using similar analysis as that in [24].

**Monotonicity.** We present the definition of monotonicity as follows.  $\square$

**Definition 3.6.** (Monotonicity) The numerical scheme  $\mathcal{F}^h$  is monotone if, given  $\phi_{i,j}^n \leq \psi_{i,j}^n$  for all  $i, j, n$ , the following inequality holds:

$$\mathcal{F}^h(\mathbf{p}_{i,j}^{n+1}, \phi_{i,j}^{n+1}, \{\phi_{p,q}^{n+1}\}_{(p,q) \neq (i,j)}, \phi_{i,j}^n) \leq \mathcal{F}^h(\mathbf{p}_{i,j}^{n+1}, \psi_{i,j}^{n+1}, \{\psi_{p,q}^{n+1}\}_{(p,q) \neq (i,j)}, \psi_{i,j}^n). \tag{3.52}$$

The positive coefficient enforcement in our numerical scheme guarantees monotonicity:

**Theorem 3.7.** The numerical schemes defined in (3.5) and (3.28) are monotone according to Definition 3.6.

**Proof.** The proof is straightforward by verifying (3.5) and (3.28).

**Convergence.** We conclude this section by proving the convergence of our numerical scheme:  $\square$

**Theorem 3.8.** (Convergence) The numerical schemes defined in (3.5) and (3.28) converge to a viscosity solution.

**Proof.** By Barles-Souganidis theory, the convergence to a viscosity solution follows since Theorems 3.3, 3.5, 3.7 demonstrate that our numerical scheme is consistent, stable and monotone [15].  $\square$

#### 4. Numerical method for non-cooperative game

After introducing a consistent, stable and monotone numerical method for cooperative game, which is represented by a 2D PDE, we consider solving the system consisting of two coupled 2D PDEs (2.10). The main challenge we encounter in solving the non-cooperative game comes from the two coupled controls. We propose a method inspired by the policy iteration technique for solving coupled HJB equations.

##### 4.1. Discretization

If the coupled values in each PDE of the system are known, the numerical algorithm for solving the PDE for the cooperative game can be applied to each equation. During the discretization, we only need to note that only one emission variable is the control and the function  $F$  for both equations are different. To facilitate our discussion, we write each discretized equation in the system

$$\frac{\mathbf{V}_k^{n+1} - \mathbf{V}_k^n}{\Delta \tau} = \sup_{E_k} \{ \mathbf{L}^{E_1, E_2} \mathbf{V}_k^{n+1} + \mathbf{B}_k^{E_k} + \mathbf{F}_k^{E_k} \}, \tag{4.1}$$

as

$$(\mathbf{I} - \Delta \tau \mathbf{L}_k^{E_1, E_2}) \mathbf{V}_k^{n+1} = \Delta \tau (\mathbf{B}_k^{E_k} + \mathbf{F}_k^{E_k}) + \mathbf{V}_k^n, \tag{4.2}$$

where

$$E_k^* = \operatorname{argsup}_{E_k} \{ \mathbf{L}^{E_1, E_2} \mathbf{V}_k^{n+1} + \mathbf{B}_k^E + \mathbf{F}_k^E \}, \quad k = 1, 2. \tag{4.3}$$

The column vector  $\mathbf{V}_k^n$  consists of all unknowns in the equation for the  $k$ th region:

$$\mathbf{V}_k^n = ((V_k)_{1,1}^n, (V_k)_{2,1}^n, \dots, (V_k)_{N_p-1,1}^n, \dots, (V_k)_{1,N_s-1}^n, \dots, (V_k)_{N_p-1,N_s-1}^n)^T,$$

where  $(V_k)_{i,j}^n$  is the numerical solution to  $V_k$  at  $(P_i, S_j, \tau^n)$  and the  $l$ th entry of the above vector is

$$(V_k)_l^n = (V_k)_{i,j}^n, \quad l = i - 1 + (j - 1)(N_p - 1). \tag{4.4}$$

The vector  $\mathbf{B}_k^E$  contains the boundary values and  $\mathbf{F}_k^E$  represents the values of  $F_k$  at inner grid points.

##### 4.2. Solving the coupled problem

In the above section, we assume that the coupled control in each equation of the system is known, which allows us to solve the equation by following the numerical scheme for cooperative game. An intuitive idea is to fix one coupled control in one equation, solve this equation numerically, solve the other equation numerically using the obtained coupled control and repeat the previous steps until convergence. Similar to the policy iteration, this method iterates between  $(V_1, E_1)$  and  $(V_2, E_2)$ . We note, however, that this method introduces too much computational cost. In each time stepping, each region has to be solved repeatedly until equilibrium between them is reached. The policy iteration is used to solve (4.1) numerically, which involves computations for solving linear system multiple times. Since the controls are the only variables that are coupled within the system, we propose another method that switches the order of iteration between coupled variables. Instead of iterating between  $V_k$  and  $E_k$  (the policy iteration) and then between  $(V_1, E_1)$  and  $(V_2, E_2)$ , we combine the policy iterations for both equations by iterating between  $(V_1, V_2)$  and  $(E_1, E_2)$ . The description of this algorithm is given as follows:

**Algorithm 4** Solving the system for the non-cooperative game at the  $n$ th time step.

- 1: Let  $\hat{\mathbf{V}}_k^s, E_k^s$  be the numerical approximations to  $\mathbf{V}_k^{n+1}$  and  $(E_i^{n+1})^s$  after the  $s$ th iteration respectively, where  $k = 1, 2$ .
- 2: Let  $\hat{\mathbf{V}}_k^0 = \mathbf{V}_k^n, k = 1, 2$ .
- 3: **for**  $s = 0, 1, 2, \dots$  until convergence ( $\|\hat{\mathbf{V}}_k^{s+1} - \hat{\mathbf{V}}_k^s\|$  is small enough,  $k = 1, 2$ ) **do**
- 4: Find  $E_k^s (k = 1, 2)$  such that

$$E_1^s = \underset{E_1}{\operatorname{argsup}}\{\mathbf{L}^{E_1, E_2^s} \hat{\mathbf{V}}_1^s + \mathbf{B}_1^{E_1} + \mathbf{F}_1^{E_1}\} \tag{4.5}$$

5: and

$$E_2^s = \underset{E_2}{\operatorname{argsup}}\{\mathbf{L}^{E_1^s, E_2} \hat{\mathbf{V}}_2^s + \mathbf{B}_2^{E_2} + \mathbf{F}_2^{E_2}\}. \tag{4.6}$$

6: Obtain  $\hat{\mathbf{V}}_1^{s+1}$  and  $\hat{\mathbf{V}}_2^{s+1}$  by solving

$$(\mathbf{I} - \Delta\tau \mathbf{L}^{E_1^s, E_2^s}) \hat{\mathbf{V}}_k^{s+1} = \Delta\tau (\mathbf{B}_k^{E_k^s} + \mathbf{F}_k^{E_k^s}) + \mathbf{V}_k^n, \quad k = 1, 2. \tag{4.7}$$

7: The numerical solutions to  $\mathbf{V}_k^{n+1}$  in (4.2) and  $E_k^*$  in (4.3) are given by  $\hat{\mathbf{V}}_k^{s+1}$  and  $E_k^s$ , respectively.

In the above algorithm, Eqs. (4.5) and (4.6) are coupled with each other by the controls. The controls  $E_k^s (k = 1, 2)$  can be solved through iteration between them, which is similar to the policy iteration between the value function and the control. We note that such iteration can be avoided if we approximate the controls for discretized equations by the controls for the PDE. According to (2.10), the expression of  $E_1$  can be written as:

$$E_1 = \underset{\hat{E}_1}{\operatorname{argsup}}\{\mathcal{L}^{\hat{E}_1, E_2} V_1 + F_1(P, S, \tau, \hat{E}_1)\}. \tag{4.8}$$

Through the observation of (2.11) and (2.12), we find that  $E_2$  is irrelevant in the above equation, which means that the analytical expression of  $E_1$  can be obtained without knowledge of  $E_2$ :

$$E_1 = \underset{\hat{E}_1}{\operatorname{argsup}}\{\mathcal{L}^{\hat{E}_1} V_1 + F_1(P, S, \tau, \hat{E}_1)\} = A_1 - S + \frac{\partial V_1}{\partial P}. \tag{4.9}$$

Similarly, we have

$$E_2 = \underset{\hat{E}_2}{\operatorname{argsup}}\{\mathcal{L}^{\hat{E}_2} V_2 + F_2(P, S, \tau, \hat{E}_2)\} = A_2 - S + \frac{\partial V_2}{\partial P}. \tag{4.10}$$

The numerical approximation to the controls can be obtained by discretizing the first order derivative term with respect to  $P$ . We note that replacing the controls for discretized equations with the controls for PDE to avoid iteration is time-effective and sometimes necessary to ensure convergence. In our experiment, we find that the equilibrium between the two controls in (4.5) and (4.6) may not be reached at some grid points for reasonable small stopping criterion.

### 4.3. Convergence analysis

For the cooperative game, we propose a numerical scheme and prove its convergence to the viscosity solution. The theory of viscosity solution for systems is introduced in [25]. The concepts of monotonicity, regularity, and consistency are discussed in [26], which theoretically proves the convergence of weakly coupled systems of HJB equations for approximation schemes. It can be easily verified that our scheme satisfies the conditions (C1) to (C3) in [26]. For (C5), the existence of discrete solution for each mesh  $h$ , we proceed to prove the convergence of the iteration between those coupled variables in Algorithm 4. Based on the Proposition 3.3 in [26], our scheme for the systems converges locally uniformly.

**Theorem 4.1.** *The iteration between value functions and controls described in Algorithm 4 converges when  $h \rightarrow 0$ , if we have  $\|(\mathbf{I} - \Delta\tau \mathbf{L}^{E_1^s, E_2^s})^{-1}\|_\infty < c < 1$ .*

**Proof.** We write  $\hat{\mathbf{U}}_1^s = \hat{\mathbf{V}}_1^s + \hat{\mathbf{V}}_2^s$  and  $\hat{\mathbf{U}}_2^s = \hat{\mathbf{V}}_1^s - \hat{\mathbf{V}}_2^s$  for convenience. By (4.7), we have

$$\begin{aligned} (\mathbf{I} - \Delta\tau \mathbf{L}^{E_1^s, E_2^s})(\hat{\mathbf{V}}_1^{s+1} - \hat{\mathbf{V}}_1^s) &= \mathbf{V}_1^n + \Delta\tau (\mathbf{B}_1^{E_1^s} + \mathbf{F}_1^{E_1^s}) - \hat{\mathbf{V}}_1^s + \Delta\tau \mathbf{L}^{E_1^s, E_2^s} \hat{\mathbf{V}}_1^s \\ &= \Delta\tau (\mathbf{L}^{E_1^s, E_2^s} \hat{\mathbf{V}}_1^s + \mathbf{B}_1^{E_1^s} + \mathbf{F}_1^{E_1^s}) - \Delta\tau (\mathbf{L}^{E_1^{s-1}, E_2^{s-1}} \hat{\mathbf{V}}_1^s + \mathbf{B}_1^{E_1^{s-1}} + \mathbf{F}_1^{E_1^{s-1}}). \end{aligned} \tag{4.11}$$

Similarly, for the second region, we have

$$(\mathbf{I} - \Delta\tau \mathbf{L}^{E_1^s, E_2^s})(\hat{\mathbf{V}}_2^{s+1} - \hat{\mathbf{V}}_2^s) = \Delta\tau (\mathbf{L}^{E_1^s, E_2^s} \hat{\mathbf{V}}_2^s + \mathbf{B}_2^{E_2^s} + \mathbf{F}_2^{E_2^s}) - \Delta\tau (\mathbf{L}^{E_1^{s-1}, E_2^{s-1}} \hat{\mathbf{V}}_2^s + \mathbf{B}_2^{E_2^{s-1}} + \mathbf{F}_2^{E_2^{s-1}}). \tag{4.12}$$

Adding (4.11) and (4.12) together, we have

$$(I - \Delta \tau \mathbf{L}^{E_1^s, E_2^s})(\hat{\mathbf{U}}_1^{s+1} - \hat{\mathbf{U}}_1^s) = \Delta \tau ((\mathbf{L}^{E_1^s, E_2^s} - \mathbf{L}^{E_1^{s-1}, E_2^{s-1}})\hat{\mathbf{U}}_1^s + \mathbf{B}_1^{E_s} + \mathbf{F}_1^{E_s} + \mathbf{B}_2^{E_s} + \mathbf{F}_2^{E_s} - \mathbf{B}_1^{E_s-1} - \mathbf{F}_1^{E_s-1} - \mathbf{B}_2^{E_s-1} - \mathbf{F}_2^{E_s-1}). \tag{4.13}$$

We note that it is only the discretization of  $\frac{\partial V_k}{\partial p}$  that makes  $\mathbf{L}^{E_1^s, E_2^s}$  and  $\mathbf{L}^{E_1^{s-1}, E_2^{s-1}}$ ,  $\mathbf{B}_k^{E_s}$  and  $\mathbf{F}_k^{E_s}$  different. We write the  $l$ th row of the column vector  $(\mathbf{L}^{E_1^s, E_2^s} - \mathbf{L}^{E_1^{s-1}, E_2^{s-1}})\hat{\mathbf{U}}_1^s + \mathbf{B}_1^{E_s} + \mathbf{B}_2^{E_s} - \mathbf{B}_1^{E_s-1} - \mathbf{B}_2^{E_s-1}$  as

$$((E_1^s)_{i,j} + (E_2^s)_{i,j} - \theta_p P_i) D_{i,j}^s(\hat{\mathbf{U}}_1^s) - ((E_1^{s-1})_{i,j} + (E_2^{s-1})_{i,j} - \theta_p P_i) D_{i,j}^{s-1}(\hat{\mathbf{U}}_1^s), \tag{4.14}$$

where  $D_{i,j}^s(\hat{\mathbf{U}}_1^s)$  stands for the discretization of the first order derivative  $\frac{\partial (V_1 + V_2)}{\partial p}$  at  $(P_i, S_j)$  when the controls are  $(E_1^s, E_2^s)$ . As  $D_{i,j}^s(\hat{\mathbf{U}}_1^s) - D_{i,j}^{s-1}(\hat{\mathbf{U}}_1^s) = O(h)$ , the above equation becomes

$$\begin{aligned} & ((E_1^s)_{i,j} + (E_2^s)_{i,j} - (E_1^{s-1})_{i,j} - (E_2^{s-1})_{i,j}) D_{i,j}^s(\hat{\mathbf{U}}_1^s) + O(h) \\ &= (D_{i,j}^s(\hat{\mathbf{U}}_1^s) - D_{i,j}^s(\hat{\mathbf{U}}_1^{s-1})) D_{i,j}^s(\hat{\mathbf{U}}_1^s) + O(h) \\ &= D_{i,j}^s(\hat{\mathbf{U}}_1^s) \mathbf{d}_{i,j}(\hat{\mathbf{U}}_1^s - \hat{\mathbf{U}}_1^{s-1}) + O(h), \end{aligned} \tag{4.15}$$

where  $\mathbf{d}_{i,j}$  is the row vector representation of the linear operator  $D_{i,j}^s$  and satisfies the requirements of row vector of  $\mathbf{J}$ . Note that the boundary condition values are eliminated due to the same discretization scheme applied to  $\hat{\mathbf{U}}_1^s$  and  $\hat{\mathbf{U}}_1^{s-1}$  and their subtraction. We now can write the vector  $(\mathbf{L}^{E_1^s, E_2^s} - \mathbf{L}^{E_1^{s-1}, E_2^{s-1}})\hat{\mathbf{U}}_1^s + \mathbf{B}_1^{E_s} + \mathbf{B}_2^{E_s} - \mathbf{B}_1^{E_s-1} - \mathbf{B}_2^{E_s-1}$  as

$$\mathbf{K}_i \mathbf{J}(\hat{\mathbf{U}}_1^s - \hat{\mathbf{U}}_1^{s-1}) + O(h), \tag{4.16}$$

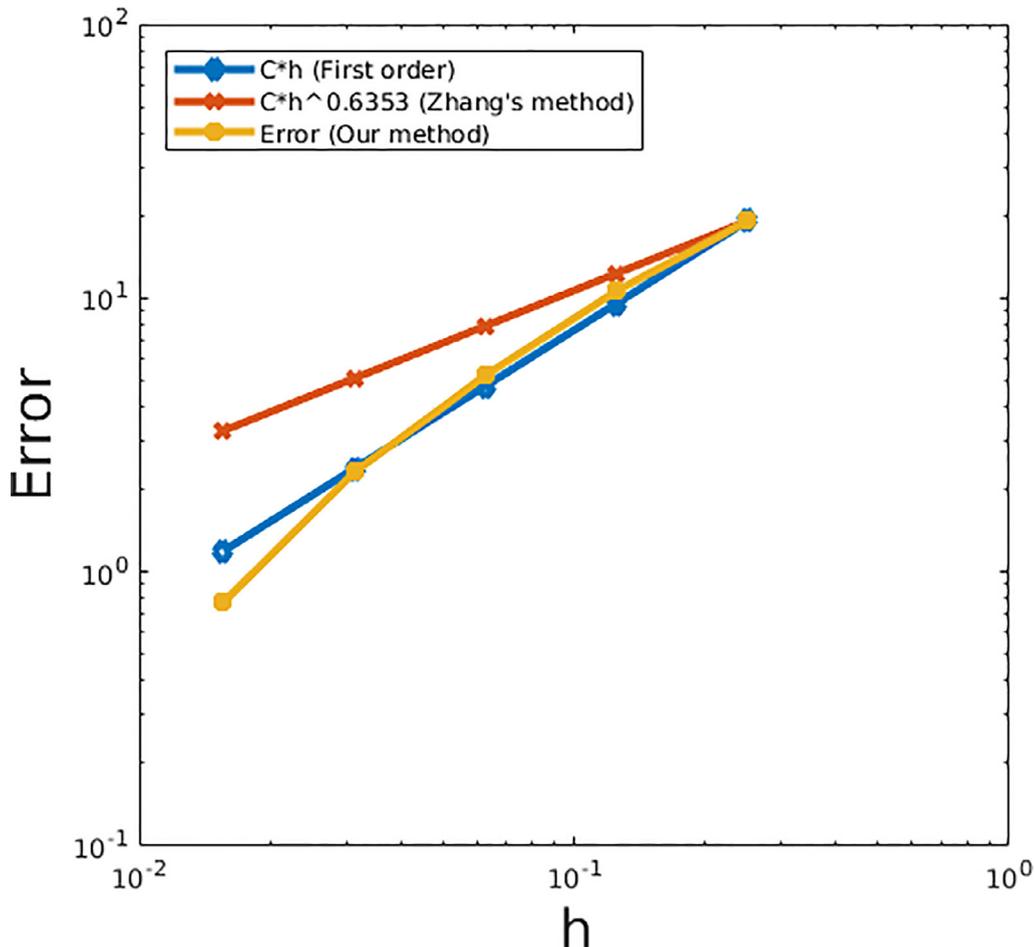


Fig. 5.1. Convergence tests on the HJB equation from the cooperative game using the wide stencil from  $n = 4$  to 8 where the solution from  $n = 9$  is treated as the exact solution. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where  $\mathbf{K}_1$  is a bounded diagonal matrix whose  $(i, j)$ -th diagonal entry is  $D_{i,j}^s(\hat{\mathbf{U}}_1^s)$  and  $\mathbf{J}$  is a matrix composed of row vectors  $\{\mathbf{d}_{i,j}\}$ . Note that  $\|\mathbf{J}\|_\infty \leq 1$  since the nonzeros in each row  $(\mathbf{J}_{l,l-1}, \mathbf{J}_{l,l}, \mathbf{J}_{l,l+1})$  is of the three options only:  $(-1, 1, 0)$ ,  $(-\frac{1}{2}, 0, \frac{1}{2})$  and  $(0, -1, 1)$ .

According to the expression of  $F_k$ , the  $l$ th row of the column vector  $\mathbf{F}_1^{E_1^s} + \mathbf{F}_2^{E_2^s} - \mathbf{F}_1^{E_1^{s-1}} - \mathbf{F}_2^{E_2^{s-1}}$  can be written as follows:

$$\begin{aligned} & \sum_{k=1,2} C_{i,j}^{s,k} ((E_k^s)_{i,j} - (E_k^{s-1})_{i,j}) \\ &= \sum_{k=1,2} C_{i,j}^{s,k} D_{i,j}^s (\hat{\mathbf{V}}_k^s - \hat{\mathbf{V}}_k^{s-1}) + O(h) \\ &= \sum_{k=1,2} C_{i,j}^{s,k} \mathbf{d}_{i,j} (\hat{\mathbf{V}}_k^s - \hat{\mathbf{V}}_k^{s-1}) + O(h), \end{aligned} \tag{4.17}$$

where  $C_{i,j}^{s,k} = A_k - S_i + \frac{1}{2}((E_k^s)_{i,j} + (E_k^{s-1})_{i,j})$ . Based on (4.17) and the definition of  $\hat{\mathbf{U}}_k^s$ , we have:

$$\begin{aligned} \mathbf{F}_1^{E_1^s} + \mathbf{F}_2^{E_2^s} - \mathbf{F}_1^{E_1^{s-1}} - \mathbf{F}_2^{E_2^{s-1}} &= \sum_{k=1,2} \mathbf{C}_k \mathbf{J} (\hat{\mathbf{V}}_k^s - \hat{\mathbf{V}}_k^{s-1}) + O(h) \\ &= \frac{1}{2}(\mathbf{C}_1 + \mathbf{C}_2) \mathbf{J} (\hat{\mathbf{U}}_1^s - \hat{\mathbf{U}}_1^{s-1}) + \frac{1}{2}(\mathbf{C}_1 - \mathbf{C}_2) \mathbf{J} (\hat{\mathbf{U}}_2^s - \hat{\mathbf{U}}_2^{s-1}) + O(h), \end{aligned} \tag{4.18}$$

where  $\mathbf{C}_k$  ( $k = 1, 2$ ) is a bounded diagonal matrix consisting of  $\{C_{i,j}^{s,k}\}$ . We write (4.18), for simplicity, as:

$$\mathbf{F}_1^{E_1^s} + \mathbf{F}_2^{E_2^s} - \mathbf{F}_1^{E_1^{s-1}} - \mathbf{F}_2^{E_2^{s-1}} = \mathbf{G}_1 \mathbf{J} (\hat{\mathbf{U}}_1^s - \hat{\mathbf{U}}_1^{s-1}) + \mathbf{G}_2 \mathbf{J} (\hat{\mathbf{U}}_2^s - \hat{\mathbf{U}}_2^{s-1}) + O(h), \tag{4.19}$$

where  $\mathbf{G}_1 = \frac{1}{2}(\mathbf{C}_1 + \mathbf{C}_2)$  and  $\mathbf{G}_2 = \frac{1}{2}(\mathbf{C}_1 - \mathbf{C}_2)$ . Substituting (4.17) and (4.19) into (4.13), we obtain

$$(\mathbf{I} - \Delta \tau \mathbf{L}^{E_1^s, E_2^s}) (\hat{\mathbf{U}}_1^{s+1} - \hat{\mathbf{U}}_1^s) = \Delta \tau (\mathbf{K}_1 + \mathbf{G}_1) \mathbf{J} (\hat{\mathbf{U}}_1^s - \hat{\mathbf{U}}_1^{s-1}) + \mathbf{G}_2 \mathbf{J} (\hat{\mathbf{U}}_2^s - \hat{\mathbf{U}}_2^{s-1}) + O(h^2). \tag{4.20}$$

Let the constant  $M = \max(\|\mathbf{K}_1 + \mathbf{G}_1\|_\infty, \|\mathbf{G}_2\|_\infty)$ . Then the infinity norm of  $\hat{\mathbf{U}}_1^{s+1} - \hat{\mathbf{U}}_1^s$  becomes

$$\|\hat{\mathbf{U}}_1^{s+1} - \hat{\mathbf{U}}_1^s\|_\infty \leq \Delta \tau M \|(\mathbf{I} - \Delta \tau \mathbf{L}^{E_1^s, E_2^s})^{-1} \mathbf{J}\|_\infty (\|\hat{\mathbf{U}}_1^s - \hat{\mathbf{U}}_1^{s-1}\|_\infty + \|\hat{\mathbf{U}}_2^s - \hat{\mathbf{U}}_2^{s-1}\|_\infty) + O(h^2). \tag{4.21}$$

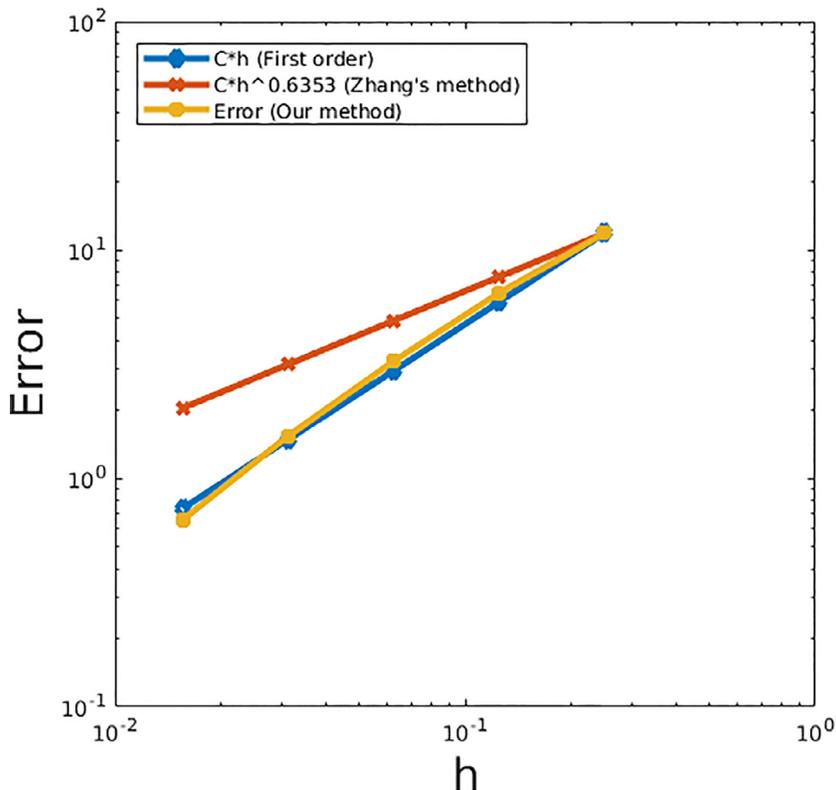


Fig. 5.2. Convergence tests for system from the non-cooperative game using the narrow stencil only from  $n = 4$  to 8 where the solution from  $n = 10$  is treated as the exact solution.

Similarly, we have

$$\|\hat{\mathbf{U}}_2^{s+1} - \hat{\mathbf{U}}_2^s\|_\infty \leq \Delta\tau M \|(\mathbf{I} - \Delta\tau \mathbf{L}^{E_1^s, E_2^s})^{-1} \mathbf{J}\|_\infty (\|\hat{\mathbf{U}}_1^s - \hat{\mathbf{U}}_1^{s-1}\|_\infty + \|\hat{\mathbf{U}}_2^s - \hat{\mathbf{U}}_2^{s-1}\|_\infty) + O(h^2). \tag{4.22}$$

Applying the condition that  $\|(\mathbf{I} - \Delta\tau \mathbf{L}^{E_1^s, E_2^s})^{-1}\|_\infty < c < 1$ , we can easily obtain  $\|(\mathbf{I} - \Delta\tau \mathbf{L}^{E_1^s, E_2^s})^{-1} \mathbf{J}\|_\infty \leq \|(\mathbf{I} - \Delta\tau \mathbf{L}^{E_1^s, E_2^s})^{-1}\|_\infty \|\mathbf{J}\|_\infty < c < 1$ . When  $s \rightarrow \infty$  we have

$$\|\hat{\mathbf{U}}_1^{s+1} - \hat{\mathbf{U}}_1^s\|_\infty = O(h^2), \tag{4.23}$$

and

$$\|\hat{\mathbf{U}}_2^{s+1} - \hat{\mathbf{U}}_2^s\|_\infty = O(h^2). \tag{4.24}$$

The above two equations together imply

$$\|\hat{\mathbf{V}}_k^{s+1} - \hat{\mathbf{V}}_k^s\|_\infty = O(h^2), \quad k = 1, 2, \tag{4.25}$$

which concludes our proof.

We note that the term  $O(h^2)$  may be removed if the first order derivative discretization schemes for the  $s - 1$  and  $s$ th steps are the same when  $s \rightarrow \infty$ . In this case,  $\hat{\mathbf{V}}_k^s$  converges. As long as we set a small  $h$ , the embedded policy iteration (Algorithm 4) for non-cooperative games will converge regardless of  $h$ . The condition  $\|(\mathbf{I} - \Delta\tau \mathbf{L}^{E_1^s, E_2^s})^{-1}\|_\infty < c < 1$  can be met when we use wide stencil for the systems by conducting similar analysis in Lemma 3.4 and Theorem 3.5 on  $\mathbf{I} - \Delta\tau \mathbf{L}^{E_1^s, E_2^s}$ .  $\square$

### 5. Numerical results

We first present the numerical results for cooperative and non-cooperative games to demonstrate the first order convergence with respect to the mesh parameter  $h$ . We use the parameters chosen by [11], which is shown in Table 5.1.

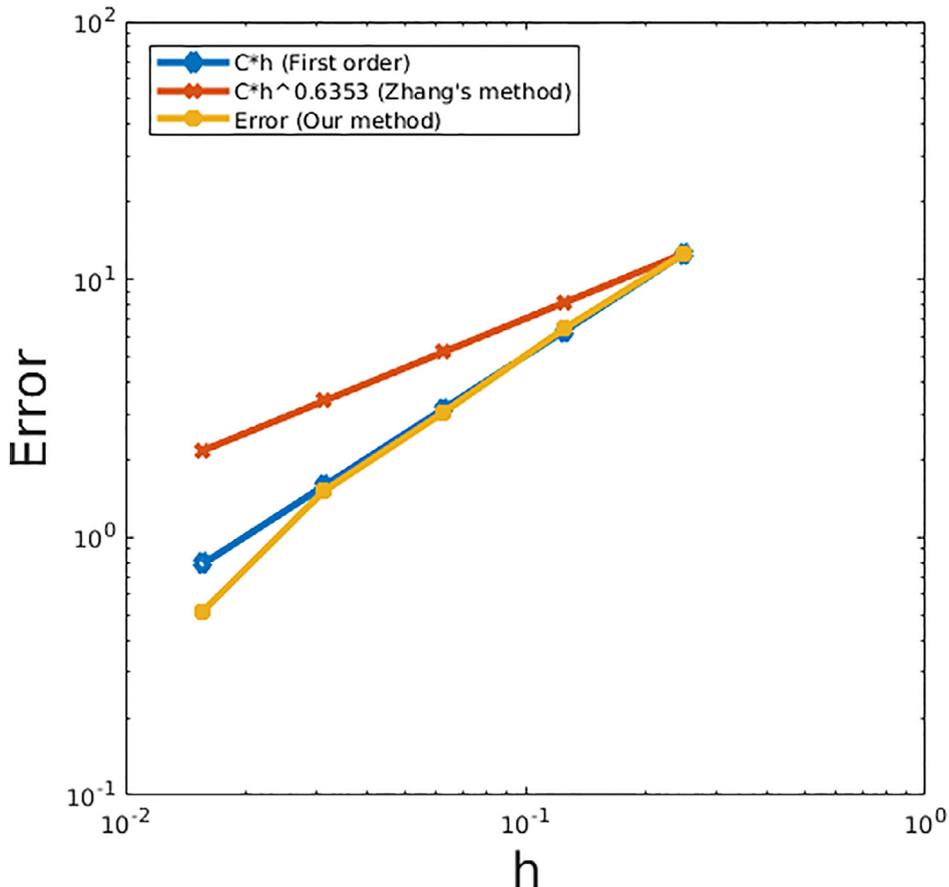


Fig. 5.3. Convergence tests for system from the non-cooperative game using the wide stencil only from  $n = 4$  to 8 where the solution from  $n = 9$  is treated as the exact solution.

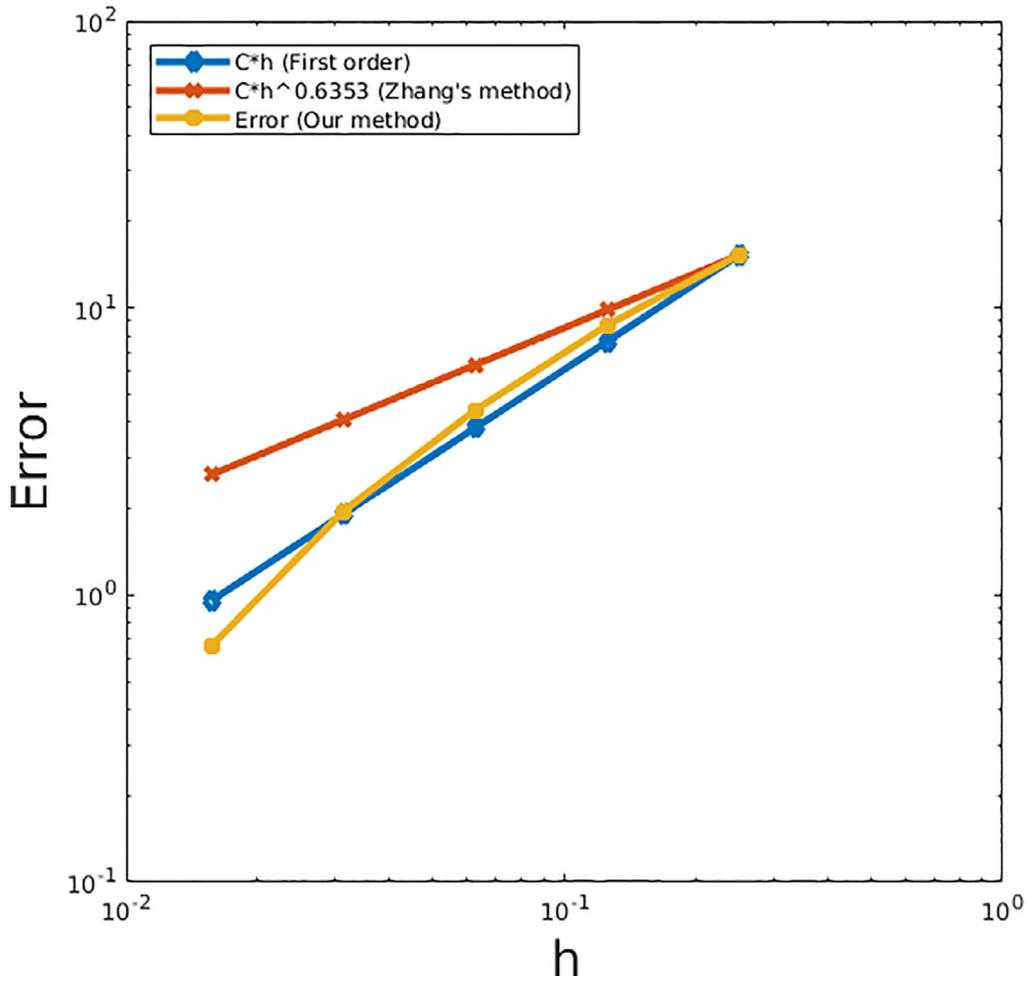


Fig. 5.4. Convergence tests on the HJB equation from the cooperative game with new parameters using the wide stencil from  $n = 4$  to 8 where the solution from  $n = 9$  is treated as the exact solution.

Table 5.1  
Parameters chosen for experiments [11].

Parameter	Value
$P_{min}$ , minimum value of pollution stock $P$	100
$P_{max}$ , maximum value of pollution stock $P$	1000
$S_{min}$ , minimum value of emission permits price $S$	0
$S_{max}$ , maximum value of emission permits price $S$	2
$T$ , maturity of the game	2
$A_1$ , parameter related to emission in region 1	5
$A_2$ , parameter related to emission in region 2	4.5
$D_1$ , parameter related to pollution stock in region 1	0.1
$D_2$ , parameter related to pollution stock in region 2	0.12
$E_{10}$ , initial emission quota for region 1	5
$E_{20}$ , initial emission quota for region 2	6
$g_1$ , parameter related to salvage cost at time $T$ in region 1	3
$g_2$ , parameter related to salvage cost at time $T$ in region 2	2
$\bar{P}_1$ , parameter related to salvage cost at time $T$ in region 1	1100
$\bar{P}_2$ , parameter related to salvage cost at time $T$ in region 2	1200
$\sigma_P$ , volatility of pollution stock	0.3
$\sigma_S$ , volatility of emission permits price	0.3
$\theta_P$ , exponential decay rate of pollution	0.3
$\mu_S$ , drift rate of emission permits price	0.3
$\rho$ , correlation coefficient	0.5
$r$ , risk-free discount rate	0.08

We choose the discretization parameters as:  $N_p = N_s = N_\tau = 2^n$  where  $n > 1$ . Note that the range of variable  $P$  is so large that the space step for  $P$  ( $\Delta P$ ) is always larger than 1. Therefore, the assumption for the wide stencil implementation ( $h < \sqrt{h}$ ) is violated. To overcome this deficiency, we apply a change of variable to the 2D equation by letting  $Q = \frac{P}{s}$  where  $s$  is a constant scaling factor. The expression of  $L^E$ , namely, (2.7) becomes:

$$\left(\frac{E_1 + E_2}{s} - \theta_P Q\right) \frac{\partial V}{\partial Q} + \frac{1}{2} \sigma_P^2 Q^2 \frac{\partial^2 V}{\partial P^2} + \mu_S S \frac{\partial V}{\partial S} + \frac{1}{2} \sigma_S^2 S^2 \frac{\partial^2 V}{\partial S^2} + \rho \sigma_P \sigma_S Q S \frac{\partial^2 V}{\partial Q \partial S} - rV + (A_1 - S)E_1 + (A_2 - S)E_2 - \frac{E_1^2 + E_2^2}{2} + (E_{10} + E_{20})S - (D_1 + D_2)SQ, \tag{5.1}$$

with the initial condition

$$\begin{aligned} V(Q, S, 0) &= -g_1(sQ - \bar{P}_1) - g_2(sQ - \bar{P}_2) \\ &= -(g_1 + g_2)sQ + g_1\bar{P}_1 + g_2\bar{P}_2. \end{aligned}$$

The above equation is slightly different from (2.7) but the same method can be applied to solve it. We note, however, that wide stencil discretization will lead to more nonzeros in the discretized matrix when it comes to solve the linear system arising from discretization.

The tests for the cooperative game are done for  $n = 4$  to 8. We treat the numerical solution for  $n = 9$  as the exact solution in order to estimate the error under the infinity norm. The results of our experiments on the cooperative model are presented in Fig. 5.1. The  $x$ -axis represents  $\frac{1}{2^n}$ , which is proportional to  $h$  while the  $y$ -axis represents error under the infinity norm. The correlation between the error from our numerical solution and  $h$  is shown by the green line. Fig. 5.1 demonstrates the first order convergence of our numerical scheme, which is superior to the result, shown as the orange line, from Chang et al. [11].

For the non-cooperative game, we use wide stencil discretization to guarantee the M-matrix for the convergence of the coupled discretized equations in Algorithm 4. We also present the results when narrow stencil is used only since the wide

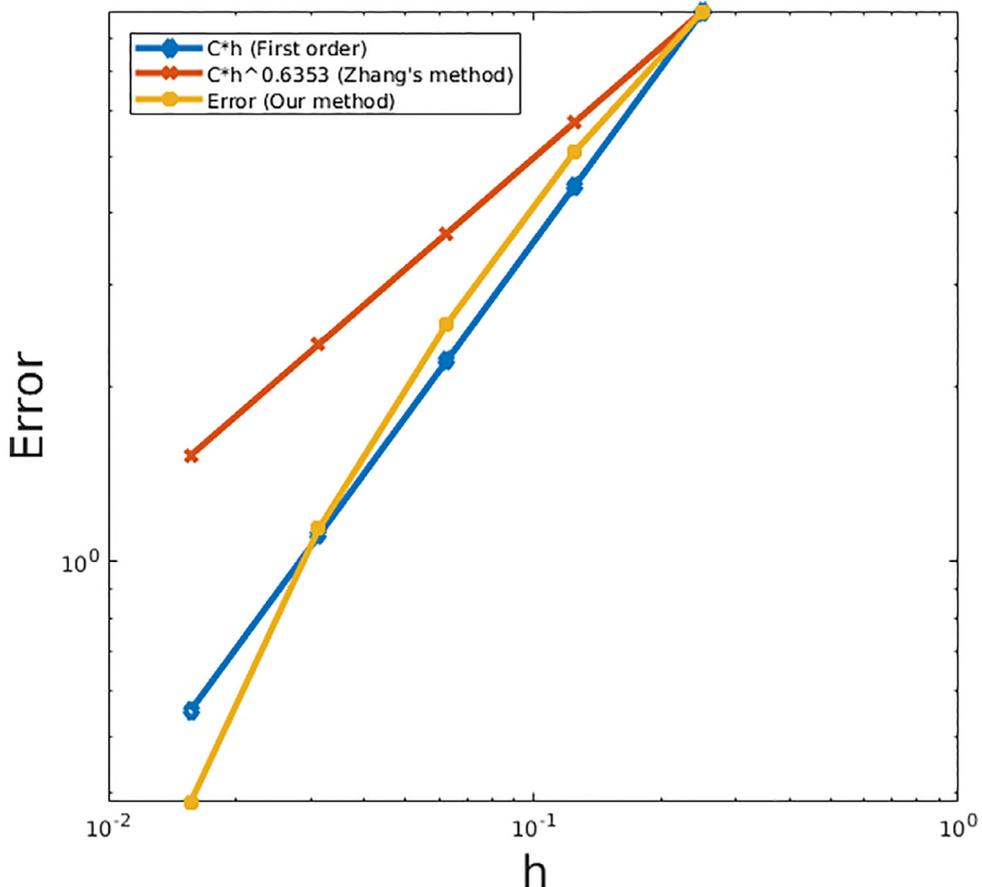


Fig. 5.5. Convergence tests for system from the non-cooperative game with new parameters using the wide stencil only from  $n = 4$  to 8 where the solution from  $n = 9$  is treated as the exact solution.

stencil takes more time to solve. Our test results show that our numerical scheme converges at almost first order for both wide and narrow stencils (Figs. 5.2 and 5.3).

We also change the parameters  $\sigma_S$  and  $\mu_S$  to 0.3 and 0.2 respectively and experiment both games using wide stencil. First order convergence is still achieved using new parameters (Figs. 5.4 and 5.5). To further strengthen our claim that our scheme has first order convergence, we try our scheme on a similar model originated from [4]. In addition to the emission, we add the pollution abatement effort as another control. We denote the pollution abatement effort at time  $t$  for the  $i$ th player as  $G_i(t)$ . Therefore, the stochastic Eq. (2.1) becomes

$$dP = (E_1 + E_2 - G_1\sqrt{P} - G_2\sqrt{P} - \theta_P P)dt + \sigma_P P dW_P. \tag{5.2}$$

The expression of net revenue for region  $i$  (2.3) is formulated as follows:

$$\mathcal{R}_i(t) = (A_i - S(t))E_i(t) - \frac{1}{2}E_i^2(t) - c_i G_i^2(t) + S(t)E_{i0} - D_i P(t), \tag{5.3}$$

where  $c_i > 0$ . The stochastic equation for the permit price remains unchanged. Applying dynamic programming principle and Ito's Lemma, we obtain new HJB equations for both cooperative and non-cooperative games. For the cooperative game, we have

$$\frac{\partial V}{\partial \tau} = \sup_{E_1, G_1, E_2, G_2} \{ \mathcal{L}^{E, G} V + F(P, S, \tau, E_1, G_1, E_2, G_2) \}, \tag{5.4}$$

where  $V$  is the joint value function. The expression of  $\mathcal{L}^{E, G} V$  is given as follows:

$$\left( \sum_{i=1}^2 (E_i - G_i\sqrt{P}) - \theta_P P \right) \frac{\partial V}{\partial P} + \frac{1}{2} \sigma_P^2 P^2 \frac{\partial^2 V}{\partial P^2} + \mu_S S \frac{\partial V}{\partial S} + \frac{1}{2} \sigma_S^2 S^2 \frac{\partial^2 V}{\partial S^2} + \rho \sigma_P \sigma_S P S \frac{\partial^2 V}{\partial P \partial S} - rV, \tag{5.5}$$

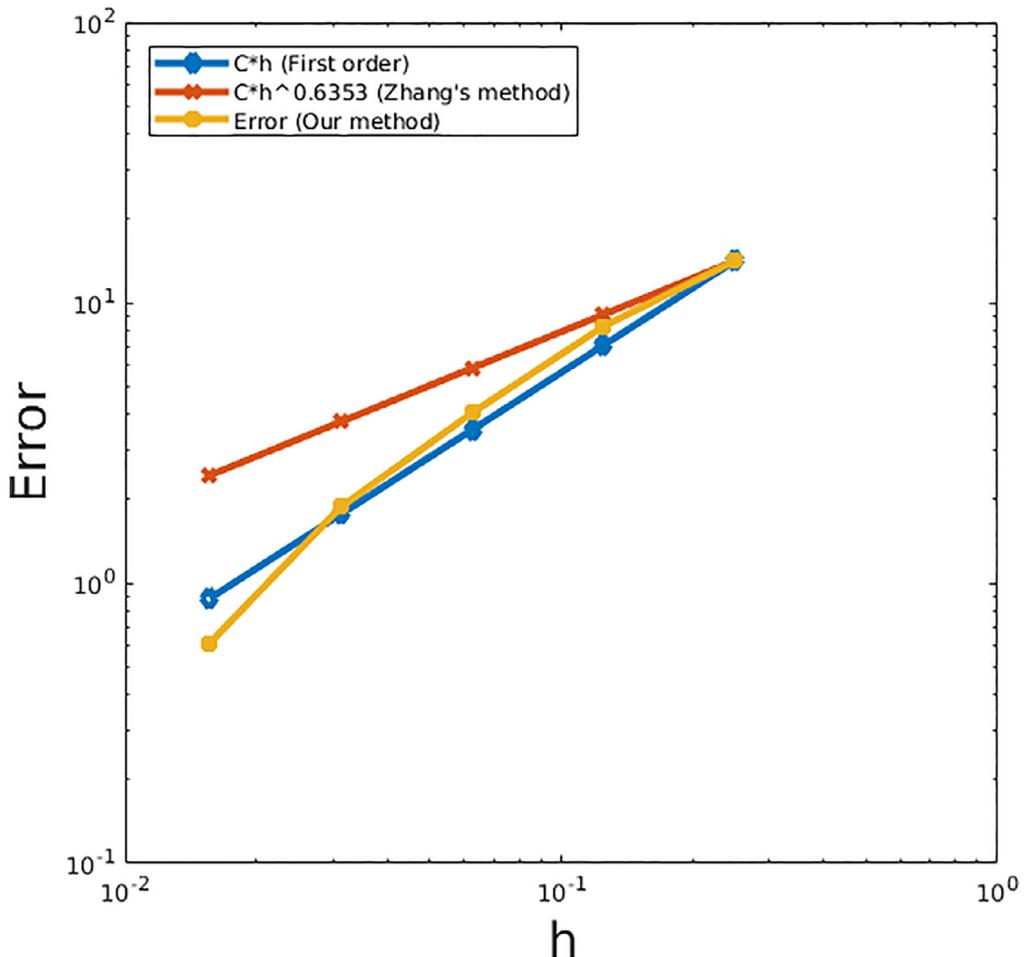


Fig. 5.6. Convergence tests on the HJB equation from the cooperative game of the new example using the wide stencil from  $n = 4$  to 8 where the solution from  $n = 9$  is treated as the exact solution.

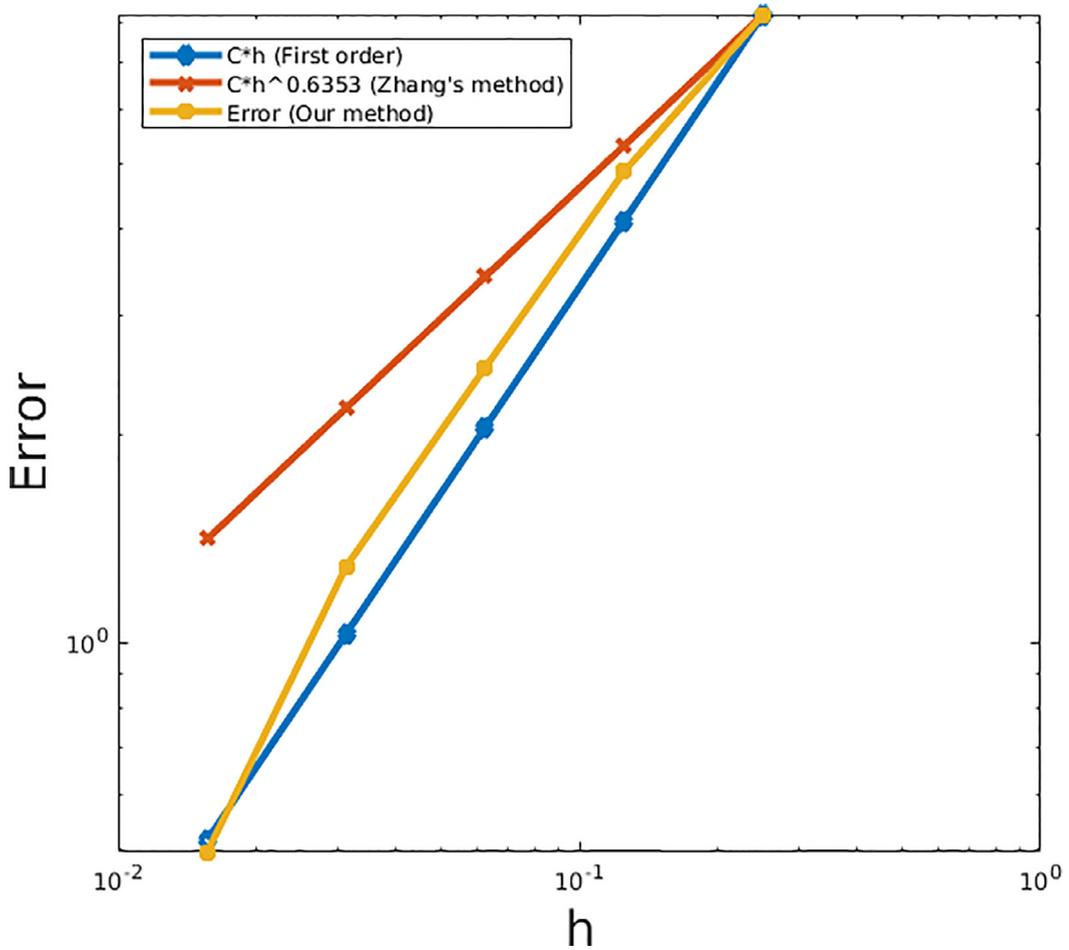


Fig. 5.7. Convergence tests for system from the non-cooperative game of the new example using the wide stencil only from  $n = 4$  to 8 where the solution from  $n = 9$  is treated as the exact solution.

and

$$F(P, S, \tau, E_1, G_1, E_2, G_2) = \sum_{i=1}^2 \left( (A_i - S)E_i - \frac{E_i^2}{2} - c_i G_i^2 + E_{i0}S - D_i P \right), \tag{5.6}$$

where the initial condition remains the same as (2.9). The HJB system for the non-cooperative game is given as follows:

$$\begin{cases} \frac{\partial V_1}{\partial \tau} = \sup_{E_1, G_1} \{ \mathcal{L}^{E, G} V_1 + F_1(P, S, \tau, E_1, G_1) \} \\ \frac{\partial V_2}{\partial \tau} = \sup_{E_2, G_2} \{ \mathcal{L}^{E, G} V_2 + F_2(P, S, \tau, E_2, G_2) \}, \end{cases} \tag{5.7}$$

where  $V_i$  ( $i = 1, 2$ ) is the value function for the  $i$ th player, the linear differential operator  $\mathcal{L}^{E, G}$  on  $V_i$ , namely,  $\mathcal{L}^{E, G} V_i$ , is given as follows:

$$\left( \sum_{i=1}^2 (E_i - G_i \sqrt{P}) - \theta_P P \right) \frac{\partial V_i}{\partial P} + \frac{1}{2} \sigma_P^2 P^2 \frac{\partial^2 V_i}{\partial P^2} + \mu_S S \frac{\partial V_i}{\partial S} + \frac{1}{2} \sigma_S^2 S^2 \frac{\partial^2 V_i}{\partial S^2} + \rho \sigma_P \sigma_S P S \frac{\partial^2 V_i}{\partial P \partial S} - r V_i \tag{5.8}$$

and

$$F_i(P, S, \tau, E_i, G_i) = (A_i - S)E_i - \frac{E_i^2}{2} - c_i G_i^2 + E_{i0}S - D_i P \tag{5.9}$$

with the the same initial condition as (2.13). We use the same boundary condition as described in Section 2.3 for both games.

We choose  $c_1 = 100$ ,  $c_2 = 200$  while other parameters remain the same as Table 5.1 and run the tests for both cooperative and non-cooperative games using wide stencil. The numerical results shown in Figs. 5.6 and 5.7 demonstrate first order convergence of our scheme.

## 6. Conclusion

In this paper, we propose numerical schemes for discretizing and solving two dimensional HJB equations and systems derived from cooperative and non-cooperative games in the transboundary pollution with emission permits trading, respectively. Our numerical scheme for HJB equations from the cooperative games is fully implicit and ensures first order convergence to viscosity solutions, which is proven theoretically. Inspired by the idea of policy iteration, we solve systems of two coupled HJB equations by iterating between value functions of the two players and their emission controls. Through theoretical analysis, we find that our scheme for systems converges. Finally, we present the numerically experimental results to demonstrate that our methods, for both cooperative and non-cooperative games, achieve first order convergence rate, which is superior to the results in [11].

## Supplementary material

Supplementary material associated with this article can be found, in the online version, at [10.1016/j.amc.2018.12.053](https://doi.org/10.1016/j.amc.2018.12.053)

## References

- [1] B.R. Copeland, M.S. Taylor, Trade and transboundary pollution, *Am. Econ. Rev.* (1995) 716–737.
- [2] M. Grubb, C. Vrolijk, D. Brack, *The Kyoto Protocol: A Guide and Assessment*, Royal Institute of International Affairs Energy and Environmental Programme, 1997.
- [3] J.L. Roelofs, United States-Canada air quality agreement: a framework for addressing transboundary air pollution problems, *Cornell Int. L. J.* 26 (1993) 421.
- [4] D.W. Yeung, Dynamically consistent cooperative solution in a differential game of transboundary industrial pollution, *J. Optim. Theory Appl.* 134 (1) (2007) 143–160.
- [5] D.W. Yeung, L.A. Petrosyan, A cooperative stochastic differential game of transboundary industrial pollution, *Automatica* 44 (6) (2008) 1532–1544.
- [6] S. Li, A differential game of transboundary industrial pollution with emission permits trading, *J. Optim. Theory Appl.* 163 (2) (2014) 642–659.
- [7] K. Chang, S.S. Wang, K. Peng, Mean reversion of stochastic convenience yields for CO<sub>2</sub> emissions allowances: empirical evidence from the EU ETS, *Span. Rev. Financ. Econ.* 11 (1) (2013) 39–45.
- [8] G. Daskalakis, D. Psychoyios, R.N. Markellos, Modeling CO<sub>2</sub> emission allowance prices and derivatives: evidence from the European trading scheme, *J. Bank. Finance* 33 (7) (2009) 1230–1241.
- [9] S. Hitzemann, M. Uhrig-Homburg, Empirical performance of reduced-form models for emission permit prices (2016). <https://ssrn.com/abstract=2297121>
- [10] J. Seifert, M. Uhrig-Homburg, M. Wagner, Dynamic behavior of CO<sub>2</sub> spot prices, *J. Environ. Econ. Manag.* 56 (2) (2008) 180–194.
- [11] S. Chang, X. Wang, Z. Wang, Modeling and computation of transboundary industrial pollution with emission permits trading by stochastic differential game, *PloS One* 10 (9) (2015) e0138641.
- [12] S. Wang, F. Gao, K. Teo, An upwind finite-difference method for the approximation of viscosity solutions to Hamilton–Jacobi–Bellman equations, *IMA J. Math. Control Inf.* 17 (2) (2000) 167–178.
- [13] F. Milner, E.-J. Park, Mixed finite-element methods for Hamilton–Jacobi–Bellman-type equations, *IMA J. Numer. Anal.* 16 (3) (1996).
- [14] M.G. Crandall, H. Ishii, P.-L. Lions, Users guide to viscosity solutions of second order partial differential equations, *Bull. Am. Math. Soc.* 27 (1) (1992) 1–67.
- [15] G. Barles, P.E. Souganidis, Convergence of approximation schemes for fully nonlinear second order equations, *Asympt. Anal.* 4 (3) (1991) 271–283.
- [16] D.M. Pooley, P.A. Forsyth, K.R. Vetzal, Numerical convergence properties of option pricing PDEs with uncertain volatility, *IMA J. Numer. Anal.* 23 (2) (2003) 241–267.
- [17] D.M. Pooley, P.A. Forsyth, K.R. Vetzal, Two factor option pricing with uncertain volatility, in: *Proceedings of the International Conference on Computational Science and Its Applications*, Springer, 2003, pp. 158–167.
- [18] S.S. Clift, P.A. Forsyth, Numerical solution of two asset jump diffusion models for option valuation, *Appl. Numer. Math.* 58 (6) (2008) 743–782.
- [19] S. Amarala, J.W. Wan, Numerical methods for dynamic Bertrand oligopoly and American options under regime switching, *Quant. Finance* 16 (11) (2016) 1741–1762.
- [20] K. Debrabant, E. Jakobsen, Semi-Lagrangian schemes for linear and fully non-linear diffusion equations, *Math. Comput.* 82 (283) (2013) 1433–1462.
- [21] K. Ma, P. Forsyth, An unconditionally monotone numerical scheme for the two-factor uncertain volatility model, *IMA J. Numer. Anal.* 37 (2) (2016) drw025.
- [22] S.S. Clift, P.A. Forsyth, Numerical solution of two asset jump diffusion models for option valuation, *Appl. Numer. Math.* 58 (6) (2008) 743–782.
- [23] P.A. Forsyth, G. Labahn, Numerical methods for controlled Hamilton–Jacobi–Bellman PDEs in finance, *J. Comput. Finance* 11 (2) (2007) 1.
- [24] Y. dHalluin, P.A. Forsyth, G. Labahn, A penalty method for American options with jump diffusion processes, *Numer. Math.* 97 (2) (2004) 321–352.
- [25] H. Ishii, S. Koike, Viscosity solutions for monotone systems of second-order elliptic PDEs, *Commun. Part. Differ. Equ.* 16 (6–7) (1991) 1095–1128.
- [26] A. Briani, F. Camilli, H. Zidani, Approximation schemes for monotone systems of nonlinear second order partial differential equations: convergence result and error estimate, *Differ. Equ. Appl.* 4 (2012) 297–317.