# Filippov Theory and Infinitesimal $\epsilon$-Greedy Q-Learning

Jack Kennedy
01351554
Supervised by Prof. S. van Strien

June 2021

## Contents

# 1 Introduction

Reinforcement learning is a form of machine learning in which an agent is given some notion of a performance score which it seeks to optimise over successive games according to some algorithm. Introduced in 1989 [1] by Watkins in an attempt to understand animal psychology, Q-learning allows for possibility of different states in which an agent may find itself and must react to. $\epsilon$-greedy Q-learning is a method for choosing actions from the performance score that will nearly always select the "best" option given the current information, but still allows for the small $\epsilon$ chance of exploration to gather more information about the game so that one is not stuck in a local maximum. Although the chance of exploration is low, the inclusion of this decision rule makes the dynamics much richer and more complex.

In this paper, I will outline the behaviour of the $\epsilon$-greedy algorithm and explore some possible solution concepts from Filippov theory for the discontinuous system of differential equations that arise from letting the learning rate go to zero, as well as numerical simulations of these different concepts and develop a discrete version of the solution concept described in [35]. In particular I discuss how letting the error of these approximation methods go to zero may not always give a single solution. This means that the question of what the Euler method with time step $h$, discussed in Section 4.1, converges to as $h \to 0$ is ill-posed for discontinuous vector fields as we can have many solutions for a given system as shown in Section 5.6.

# 2 Q-Learning

Q-learning has had many applications, from natural language processing, to training self-driving cars, to giving custom healthcare treatments. Formulated as Markov Decision Processes (MDPs) [2], they aim to solve control problems; to achieve a policy which maximises expected future returns for an agent. Here a policy is a probability distribution over the possible actions, conditioned on the state they are currently in. We will only consider single state setups in which we

have a known reward function $R$ for each iteration of the game that is played. Q-learning has two integral ingredients: an update function $u$ which determines how Q-values evolve, and a decision function $d$ which gives a distribution on the set of possible actions.

Once we have chosen these, the learning algorithm can be split up into 5 steps:

For a set of $K > 0$ players and a set $S = \{1, ..., n\}$ of $n$ different actions each agent can take on every turn :

1. Each player $k$ gets a vector $Q_k(t)$. Each component $(Q_k(t))_i$ will be a "rating" of action $i \in S$ for player $k$ before playing a game at time $t$.

2. Each player chooses/samples an action from a probability distribution on $S$ depending on the current $Q$-values according to some decision function, $a_k \sim d(Q_k(t)) \in \triangle(S)$.

3. Now all players have their action $a_k$, the game is "played" with each player taking their turn concurrently, and all players receive a reward dependent on their actions and the actions of their opponents, $R_k(a_1, ..., a_n)$.

4. All players then update their $Q$ values according to some update function $Q_k(t + h) = u(Q_k(t))$, dependent on their received reward.

5. Return to step 1 with $Q_k(t) = Q_k(t + h)$.

Here $\Delta(S)$ denotes the space of probability distributions on $S$.

## 2.1 Update Functions

In classical Q-learning, we choose the following update function:

**Definition 2.1** (Bellman Update Function)**.**

$$u : \mathbb{R}^n \times S \to \mathbb{R}^n : Q_k(t) \mapsto Q_k(t) + e_a h(R_k(a_1, ..., a_n) + \gamma \max_j (Q_k(t))_j - Q_k(t))$$

where $e_a$ is vector s.t $(e_a)_i = \delta_{ia}$, $h \in (0, 1)$ is the learning rate and $\gamma \in (0, 1)$ is the discount factor.

This complicated expression just says that each player only updates the $Q$-value for the action that player chose on that turn. The new value is the previous $Q$-value plus the temporal difference term that measures the change from the old value to the received reward plus the estimated future reward, all moderated by the learning rate $h$. The reward is also usually held constant in time, given by a matrix so that each combination of actions by all the players gives a set reward to each player, regardless of how many games they have previously played. We are interested in the continuous time limit: by taking $h \to 0$, we will have the Q-values updating continuously in time, which gives their evolution as differential equations.

Single agent Q-learning ($K = 1$) is very well understood in general. Due to the fact that the single agent does not have to react to another agent/player, the environment stays fixed and therefore the evolution of the Q-values is Markovian, i.e. the policy change depends only on the current and known states of the world. Thus, given some reasonable assumptions on the learning rate and the decision rule that guarantees that every state action pair $(s, a)$ is visited at least once, the Q-values converge in measure to the optimal policy $Q^*$ which maximises the reward for the agent [3, 4].

Multi-agent Q-learning is much more complicated, as the optimal behaviour each player is trying to learn depends on the behaviour of the other players, which are also changing according to Q-learning. Multi-agent learning is thus inherently more complex, and is not as well studied despite its many links to evolutionary game theory and applications ranging from swarm robotics to the prediction of political policies and auctions [5, 6, 7].

From here, we shall only consider two player games with no foresight from here on i.e. $K = 2$ and $\gamma = 0$. We will also require that our game is normal form and that the reward is constant in time: if player 1 and 2 play pure actions $a$ and $b$ respectively, the reward to player 1 and player 2 is given by

$$R_A^t(a, b) = e_a.Ae_b \text{ and } R_B^t(a, b) = e_b.Be_a$$

respectively, for some $2 \times 2$ matrices $A$ and $B$. If players play mixed actions (i.e a distribution over $S$) $a$ and $b$, where $a_i$ is the probability that $A$ plays action $i$ and likewise for $b$, the expected reward to $A$ will be

$$\sum_{i=1}^{n}\sum_{j=1}^{n} P(a = a_i, b = b_j)R_A(a, b) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_i b_j e_i.Ae_j = a.Ab$$

and likewise the expected reward to $B$ will be $b.Ba$. Thus we can let our actions be replaced by mixed strategies, selecting actions according to a policy distribution over possible actions according to the decision rule, and our update function will remain unaffected.

## 2.2 Decision Functions

We now need to choose a decision function $d : \mathbb{R}^n \to \triangle(S)$ that will select an action (or a probability distribution over possible actions) from the Q-values. We should seek to find something that allows players to continue to choose actions that were beneficial in the past, but still allow for exploration of different actions so that players are not stuck in local maxima. This trade-off between exploration and exploitation is one of the great challenges of reinforcement learning. Some decision function candidates are outlined below.

**Definition 2.2.** Blind Exploration

$$? : \mathbb{R}^n \to \triangle(S) : Q \mapsto \mathrm{Uni}(S)$$

i.e. each action is chosen with equal likelihood, regardless of Q-values.

**Definition 2.3.** Greedy Exploitation

$$! : \mathbb{R}^n \to \triangle(S) : Q \mapsto \delta(\mathrm{argmax}_j\, Q_j)$$

i.e. the action with the highest Q-value is always chosen (called the "greedy" action). $\delta(x)$ is the distribution that selects $x$ with probability 1.

Note that ! is not always well-defined: if two actions have the same Q-value that are higher than the rest, $\mathrm{argmax}_j\, Q_j$ is multi-valued. Usually this is resolved by choosing uniformly from these actions in a runoff. Although this occurrence is rare for small values of the learning rate $h$, it will present an integral quirk to greedy dynamics as we will see later. For now, we shall just let it be undefined and assume we only ever have a single greedy action for each player.

? allows for us to try every action equally, but is quite useless at actually learning as it does not consider the Q-values it gained from playing previous games. On the other hand ! "over-learns" from previous games, and will remain fixated on one action which happened to have good results at one point, but is not in reality the optimal action so it is very prone to being caught in local maxima.

**Definition 2.4.** Boltzmann Exploration/Softmax

$$b : \mathbb{R}^n \to \triangle(S) : Q \mapsto \left( \frac{e^{\tau(Q)_i}}{\sum_{n=1}^{N} e^{\tau(Q)_j}} \right)_{i \in S}$$

i.e. actions are chosen with probability proportional to the exponential of their Q-values multiplied by the temperature parameter $\tau \geq 0$.

$b$ is more flexible, with higher Q-valued actions being more likely to be chosen but does not completely exclude other actions. The parameter $\tau$ controls how "sharp" this distribution is; for $\tau \approx 0$ the distribution will be close to uniform over $S$ and for $\tau \gg 1$, the distribution will be concentrated around the highest Q-valued action. Indeed $b \to ?$ as $\tau \to 0$ and $b \to !$ as $\tau \to \infty$. The behaviour of Q-learning with Boltzmann exploration is well understood in multi-agent setting. For two-player two-action games ($K = n = 2$), the Boltzmann policies $x$ and $y$ of players 1 and 2 evolve (as $h \to 0$) according to [8]:

$$\begin{aligned} \frac{dx_i}{dt} &= x_i \tau((Ay)_i - x.Ay) + x_i \sum x_j \ln\left(\frac{x_j}{x_i}\right) \\ \frac{dy_i}{dt} &= y_i \tau((Bx)_i - y.Bx) + y_i \sum y_j \ln\left(\frac{y_j}{y_i}\right) \end{aligned} \tag{1}$$

where $A$ and $B$ are the payoff matrices for players 1 and 2 respectively. The equilibria of this system even sometimes coincides with the Nash equilibria [8]. The important thing to notice is that because of the continuity of the Boltzmann decision rule, we were able to attain a continuous vector field that defined the evolution of the policies of the game and the Q-values. As we shall see with the $\epsilon$-greedy decision rule, we do not always achieve a continuous vector field.

### 2.3 $\epsilon$-greedy Q-Learning

Set $\epsilon \in (0, 1)$. We can define the $\epsilon$-greedy decision rule as:

$$d_\epsilon(Q) := \begin{cases} !(Q), & \text{with probability } 1-\epsilon \\ ?(Q), & \text{with probability } \epsilon \end{cases}$$

This means that $1-\epsilon$ of the time, we choose a greedy action with the maximal $Q$-value. Only $\epsilon$ of the time do we explore blindly and select some other action. If the set of greedy actions is a singleton, with only one greedy action $\hat{a}$, we could write this as:

$$d_\epsilon(Q) := \begin{cases} \hat{a}, & \text{with probability } 1-\epsilon+\frac{\epsilon}{n} \\ j, & \text{with probability } \frac{\epsilon}{n}, \text{ where } j \neq \hat{a} \end{cases}$$

So now we have a decision rule for which we can minutely adjust how much it explores $\epsilon$ vs how much it exploits a sure thing[1] $1-\epsilon$. Detailed analyses of this algorithm are sparse (see [11, 12]), despite its ability to give better average payoffs in many iterated games than other similar algorithms [9]. The simplicity of its implementation also makes it attractive; see [10] for an application in resource allocation. Although this decision rule is only slightly more sophisticated than the purely greedy one !, the dynamical implications of this algorithm on the $Q$-values as $h \to 0$ are incredibly enriching, with no nice encapsulations of the policy evolution as in (1). In what follows, we shall derive the discontinuous differential equations that arise from $d_\epsilon$ and study the behaviour of its trajectories when we try to integrate it using the Euler method.

For a two-player game with players $A$ & $B$ (with payoff matrices of the same name) with the (singular) greedy actions at time $t$ being $\hat{a}(t)$ & $\hat{b}(t)$, we have the following difference equation for the $Q$ values of $A$ (writing $\epsilon_0 := \frac{\epsilon}{2}$ and $\epsilon_1 := 1 - \frac{\epsilon}{2}$):

$$(Q_A(t+h))_{\hat{a}(t)} = (Q_A(t))_{\hat{a}(t)} + h \cdot \epsilon_1 \left( e_{\hat{a}(t)}.A \left[ \epsilon_1 e_{\hat{b}(t)} + \epsilon_0 e_j \right] - (Q_A(t))_{\hat{a}(t)} \right)$$

$$(Q_A(t+h))_i = (Q_A(t))_i + h \cdot \epsilon_0 \left( e_i.A \left[ \epsilon_1 e_{\hat{b}(t)} + \epsilon_0 e_j \right] - (Q_A(t))_i \right)$$

[1] We could also see $\epsilon$ as controlling how often a player meant to choose the greedy action, but made an accidental mistake in choosing another action at random.

for $i \neq \hat{a}$ and $j \neq \hat{b}$. Similarly for B:

$$(Q_B(t+h))_{\hat{b}(t)} = (Q_B(t))_{\hat{b}(t)} + h \cdot \epsilon_1 \left( e_{\hat{b}(t)}.B \left[ \epsilon_1 e_{\hat{a}(t)} + \epsilon_0 e_i \right] - (Q_B(t))_{\hat{b}(t)} \right)$$

$$(Q_B(t+h))_j = (Q_B(t))_j + h \cdot \epsilon_0 \left( e_j.B \left[ \epsilon_1 e_{\hat{a}(t)} + \epsilon_0 e_i \right] - (Q_B(t))_j \right)$$

We can now let $h \to 0$, and receive the differential equations:

$$(\dot{Q}_A(t))_{\hat{a}(t)} = \epsilon_1 \left( \epsilon_1 e_{\hat{a}(t)}.Ae_{\hat{b}(t)} + \epsilon_0 e_{\hat{a}(t)}.Ae_j - (Q_A(t))_{\hat{a}(t)} \right) \tag{2}$$

$$(\dot{Q}_A(t))_i = \epsilon_0 \left( \epsilon_1 e_i.Ae_{\hat{b}(t)} + \epsilon_0 e_i.Ae_j - (Q_A(t))_i \right) \tag{3}$$

$$(\dot{Q}_B(t))_{\hat{b}(t)} = \epsilon_1 \left( \epsilon_1 e_{\hat{b}(t)}.Be_{\hat{a}(t)} + \epsilon_0 e_{\hat{b}(t)}.Be_i - (Q_B(t))_{\hat{b}(t)} \right) \tag{4}$$

$$(\dot{Q}_B(t))_j = \epsilon_0 \left( \epsilon_1 e_j.Be_{\hat{a}(t)} + \epsilon_0 e_j.Be_i - (Q_B(t))_j \right) \tag{5}$$

for $i \neq \hat{a}$ and $j \neq \hat{b}$. From here on, this system of differential equations shall be referred to as $\epsilon$-IQL ($\epsilon$-Greedy Infinitesimal Q-learning) as in [11]. Note for $\epsilon \approx 0$, $\epsilon_0 \approx 0$ and $\epsilon_1 \approx 1$, so that most of the motion happens in the direction of the greedy action, reflecting the fact that it is selected more often.

This system is highly dependent on what the greedy actions are at time $t$, and therein lies the discontinuity. Assume the greedy action for $B$ stays fixed. If action 1 is initially the greedy action for $A$ at time $t$, its Q-value $(Q_A)_1$ will follow (2) while action 2's Q-value, $(Q_A)_2$ will follow (3). If action 2 eventually becomes the greedy action and action 1 becomes non-greedy, the equations instantly flip: $(Q_A)_1$ follows (3) and $(Q_A)_2$ follows (2). Thus the governing differential equations do not vary smoothly with $Q_A$. If instead $\hat{a}(t)$ remains constant over some time period, but the greedy action $\hat{b}(t)$ for $B$ changes (say from 1 to 2), then the expressions$\epsilon_1 e_{\hat{a}(t)}.Ae_{\hat{b}(t)}$ in (2) instantly changes from $\epsilon_1 e_{\hat{a}(t)}.Ae_1$ to $\epsilon_1 e_{\hat{a}(t)}.Ae_2$ (and similarly for (3)) i.e. the governing DEs for $Q_A$ do not vary smoothly with $Q_B$.

The discontinuities in the vector field happen along the indifference lines $\{(Q_A)_1 = (Q_A)_2\}$ and $\{(Q_B)_1 = (Q_B)_2\}$. Note that equations (2 - 5) are not defined here, as the decision rule $d_\epsilon$ is not defined here. Thus it seems we have a totally pathological vector field arising from $\epsilon$-greedy Q-learning, with no easy way to describe the evolution of the Q-values when they equal each other. However, not all is lost and ways to integrate this vector field will be proposed shortly. For now, we shall see what $\epsilon$-IQL looks like for the Prisoner's Dilemma game.

# 3 $\epsilon$-IQL on the Prisoner's Dilemma Game

Prisoner's Dilemma is a simple way to model symmetric situations in which players collectively gain more from working together (cooperation), but individ-

ually benefit more by trying to act selfishly (defection). The payoff matrices are given by

$$A = \begin{pmatrix} 1 & 4 \\ 0 & 3 \end{pmatrix},\ B = \begin{pmatrix} 1 & 0 \\ 4 & 3 \end{pmatrix} = A^T$$

So if $A$ defects while $B$ cooperates, they receive rewards 4 and 0 respectively. For brevity let $(Q_A)_i \coloneqq q_{Ai}$ and $(Q_B)_j \coloneqq q_{Bj}$. We must consider when each action for each player is the greediest: when $q_{A1} > q_{A2}$ and vice versa combined with when $q_{B1} > q_{B2}$ and vice versa. Let $D_A \coloneqq q_{A2} - q_{A1}$ and $D_B \coloneqq q_{B2} - q_{B1}$. When $D_A, D_B \neq 0$ (i.e. we are not on an indifference line), we will have well-defined equations of motion given by $\epsilon$-IQL.

Say $D_A, D_B > 0$, meaning that the second action (cooperation) is the greedy action for both players. The governing differential equations in this region are:

$$\begin{aligned}
\dot{q}_{A2} &= \epsilon_1(\epsilon_1 e_2.Ae_2 + \epsilon_0 e_2.Ae_1 - q_{A2}) \\
&= \epsilon_1(\epsilon_1 \cdot 3 + \epsilon_0 \cdot 0 - q_{A2}) \\
&= \epsilon_1(3\epsilon_1 - q_{A2}) \\
\dot{q}_{A1} &= \epsilon_0(\epsilon_1 e_1.Ae_2 + \epsilon_0 e_1.Ae_1 - q_{A1}) \\
&= \epsilon_1(\epsilon_1 \cdot 4 + \epsilon_0 \cdot 1 - q_{A1}) \\
&= \epsilon_1(4\epsilon_1 + \epsilon_0 - q_{A1}) \\
\dot{q}_{B2} &= \epsilon_1(\epsilon_1 e_2.A^T e_2 + \epsilon_0 e_1.A^T e_2 - q_{B2}) \\
&= \epsilon_1(3\epsilon_1 - q_{B2}) \\
\dot{q}_{B1} &= \epsilon_0(\epsilon_1 e_2.A^T e_1 + \epsilon_0 e_1.A^T e_1 - q_{B2}) \\
&= \epsilon_0(4\epsilon_1 + \epsilon_0 - q_{B2})
\end{aligned}$$

We need another 3 sets of equations to describe the motion in the other 3 open quadrants $D_A > 0, D_B < 0$ ($A$ cooperating, $B$ defecting), $D_A < 0, D_B > 0$ ($A$ defecting, $B$ cooperating) and $D_A, D_B < 0$ ($A$ & $B$ both defecting). These are written in Fig. 1. We will introduce some (arbitrary) labelling: let the region where $D_A, D_B > 0$ be $R_4$, $D_A > 0, D_B < 0$ be $R_3$, $D_A < 0, D_B > 0$ be $R_2$ and $D_A, D_B < 0$ be $R_1$.

## 3.1 Characteristics of the Flow

We can point out some stationary points of the vector field in Fig. 1, discontinuous and riddled with gaps though it is. Let

$$C = (\epsilon_0 + 4\epsilon_1, 3\epsilon_1) = \left(4 - \frac{3\epsilon}{2}, 3 - \frac{3\epsilon}{2}\right),\ D = (\epsilon_1 + 4\epsilon_0, 3\epsilon_0) = \left(1 + \frac{3\epsilon}{2}, \frac{3\epsilon}{2}\right)$$

Note that $C_2 - C_1 = D_2 - D_1 = -1 < 0$. These will be attractive stationary points for each individual, although they may not reach it before hitting the diagonal $D_A = 0$ or $D_B = 0$, at which point the vector field for that individual

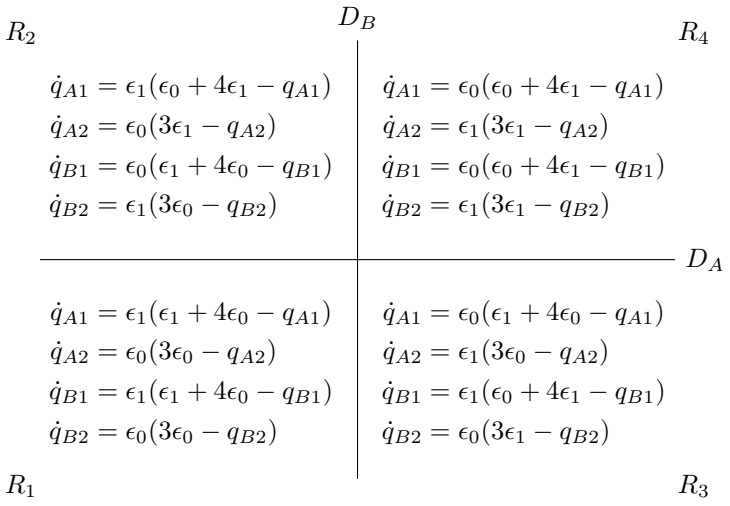


Figure 1: Governing DEs for $\epsilon$-IQL applied to the Prisoner's Dilemma game

will change and it will no longer be a stationary point. See Figures 2 & 3 for an interpretation of what these 4-dimensional vector fields looks like. The square with $C$ and $D$ as its corners is a trapping region for flows [14].

We can find tangency points on the diagonal. When both players are above the diagonal ($R_4$), the vector field is tangent to the diagonal at

$$C^\star = (C_1^\star, C_2^\star),\ C_1^\star = C_2^\star \coloneqq \frac{3\epsilon_1^2 - 4\epsilon_1\epsilon_0 - \epsilon_0^2}{\epsilon_1 - \epsilon_0} = \frac{3 - \frac{\epsilon}{2}(10 - 3\epsilon)}{1 - \epsilon} = 3 - 2\epsilon + O(\epsilon^2)$$

So on the diagonal, the vector field points away from the diagonal below $C^\star$ and points towards the diagonal above $C^\star$.

When both players are below the diagonal ($R_1$), the vector field is tangent to the diagonal at

$$D^\star = (D_1^\star, D_2^\star),\ D_1^\star = D_2^\star \coloneqq \frac{\epsilon_1^2 + 4\epsilon_1\epsilon_0 - 3\epsilon_0^2}{\epsilon_1 - \epsilon_0} = \frac{1 + \frac{\epsilon}{2}(2 - 3\epsilon)}{1 - \epsilon} = 1 + 2\epsilon + O(\epsilon^2)$$

On the diagonal, the vector field points towards the diagonal above $D^\star$ and points away below $D^\star$.

Although this vector field is discontinuous and has no solution trajectories in the classical sense, we shall attempt to integrate it anyway, in a manner that makes some mechanical sense for our problem.

# 4 Euler $\epsilon$-IQL on Prisoner's Dilemma

Due to discontinuity of the vector fields over the indifference lines, we cannot integrate to find trajectories as we usually would. One possible work around

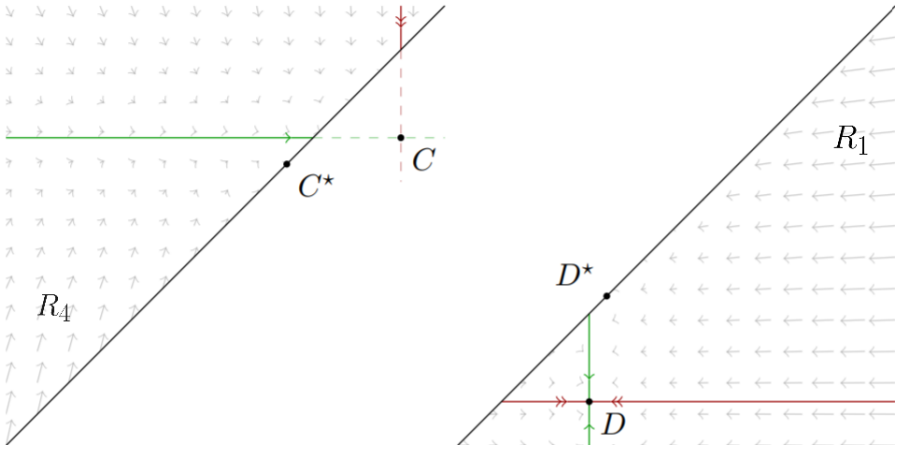


Figure 2: Illustration of the evolution of Q–values while the players have the same greedy action. On the left, $D_A, D_B > 0$ and the greedy action is to cooperate and players follow the $R_4$ vector field. In this case both players flow toward the mutual cooperation stationary point $C$ but eventually one player will hit the diagonal leading to a change in the dynamics (now following rhe $R_2$ or $R_3$ vector field). Similarly, on the right the greedy action is to defect greedily and the players follow the $R_1$ vector field. In this case both players flow toward the mutual defection stationary point $D$, which they will keep doing unless one player hits the diagonal and thus causing the dynamics to change. Taken from [14].

would be to approximate trajectories with a discrete process; since the indifference lines have measure zero, any discrete approximation will almost surely never land on them and so we can effectively ignore them. Many methods such as Runge-Kutta or exponential integrators are used to numerically integrate differential equations. In what follows, we shall follow only the (Forward) Euler method.

**Definition 4.1** (Euler Method)**.** Given the initial value problem

$$\dot{y}(t) = f(t, y(t)),\ y(t_0) = y_0 \in \mathbb{R}^m,$$

and a fixed stepsize $h > 0$, the $n$th iterate of the Forward Euler method is

$$y_n = y_{n-1} + hf(t_{n-1}, y_{n-1})$$

where $t_n \coloneqq t_0 + nh$. The Euler polygon $\eta^h$ is the piecewise linear interpolation of these points:

$$\eta^h : [t_0, \infty) \to \mathbb{R}^m : t \mapsto y_{n-1} + \frac{1}{h}(t - t_{n-1})(y_n - y_{n-1}) \text{ for } t \in [t_{n-1}, t_n]$$

When $f$ is smooth, the error for the Euler method is $O(h^2)$ and as a numerical method is usually only used as a example algorithm from which to develop more

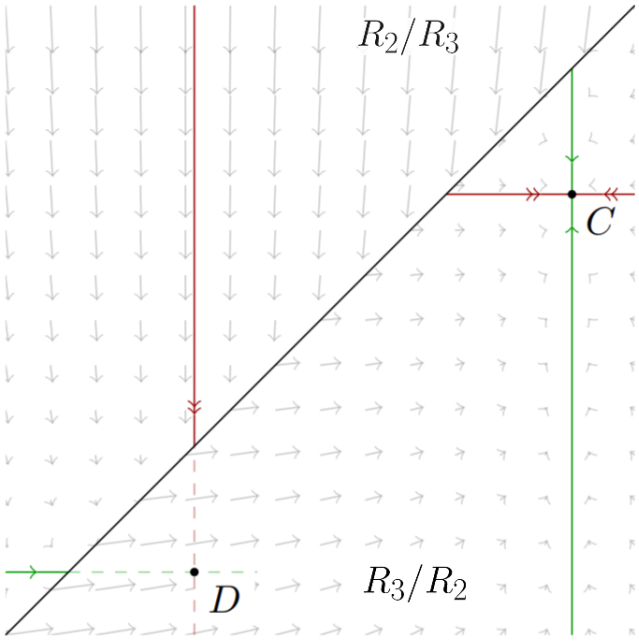


Figure 3: Illustration of the evolution of Q–values while the players have distinct greedy actions. For the player under the diagonal, defect is the greedy action and for the player above the diagonal, cooperate is the greedy action. The player under the diagonal (say $A$) flows according to the $R_3$ vector field toward the mutual cooperation stationary point C and the player above the diagonal (say $B$) flows according to the $R_2$ vector field toward the mutual defection stationary point D. Note that the player above the diagonal will eventually hit the diagonal, causing a change in the dynamics (to the vector field in $R_4$). Taken from [14].

sophisticated solvers. However, in the context of $\epsilon$-IQL, it has an appropriate mechanical interpretation and has been applied to game theoretic problems in the past [13]. Recall that $\epsilon$-IQL is Q-learning where the learning rate is taken to 0 in order to get (nearly everywhere) smooth differential equations. By using Euler method on the resultant differential equations with timestep $h$, we can interpret $h$ as the smoothed learning rate of $\epsilon$-greedy Q-learning. To recap, we had a discrete process that we made continuous, only to approximate with another discrete process. More shall be said on the Euler method and its timestep $h$ later in the context of Filippov systems. For now, we shall see what Euler $\epsilon$-IQL looks like for differential initial values, $h$s and $\epsilon$s.

## 4.1 Simulations

We start by setting the initial values slightly above the diagonal as $Q_A = (2.8, 2.801)$ and $Q_B = (2.4, 2.401)$. Let $\epsilon = 0.01$ and $h = 10^{-3}$ and see Fig-

ure 4 for the results. It would seem that after only 50 units of time, the players settle on continued mutual defection and continue in this matter (converging to $D$). Thus we must have transient behaviour and $\epsilon$-IQL has identical behaviour as in regular Q-learning.

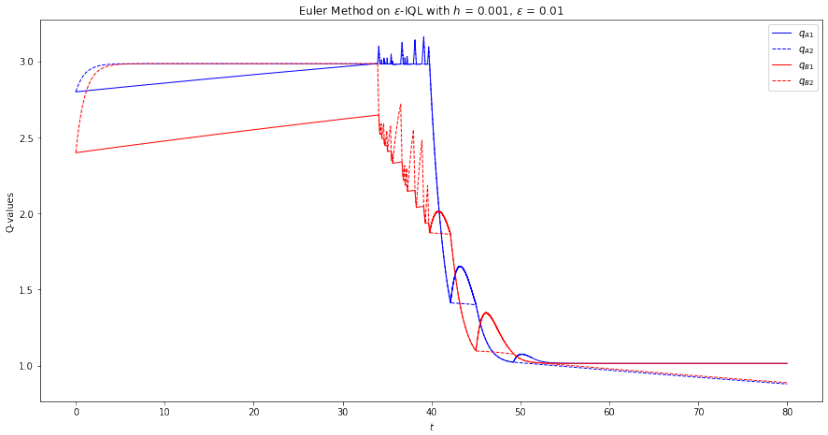


Figure 4: Transient behaviour after 50 time units

However, if we instead let $\epsilon = 0.011$ and $h = 10^{-3}$, we have behaviour as in Figure 5, where the players take more than 300 time units before settling down. If $\epsilon = 0.009$ and $h = 10^{-3}$, players do not settle to continued mutual defection until 1250 time units have passed (Figure 7). Thus the behaviour of Euler $\epsilon$-IQL depends sensitively on $\epsilon$.

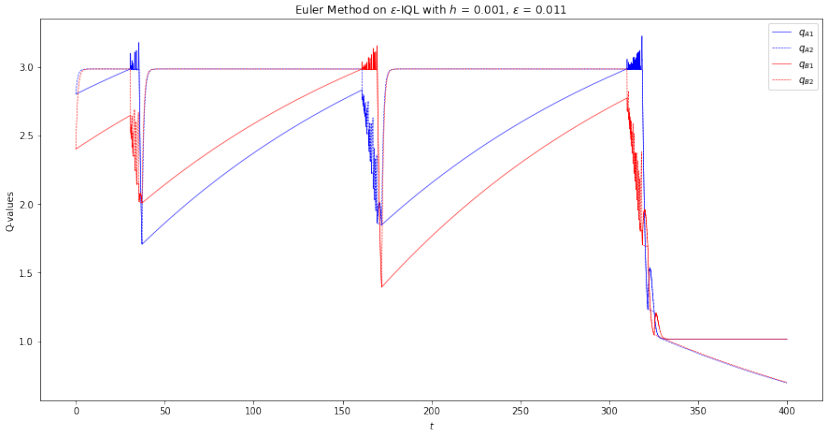


Figure 5: Transient behaviour after 300 time units

The situation becomes even more complex when varying $h$. Fix $\epsilon = 0.01$. For $h = 10^{-3}$ we know we reach continued mutual defection after 50 time units

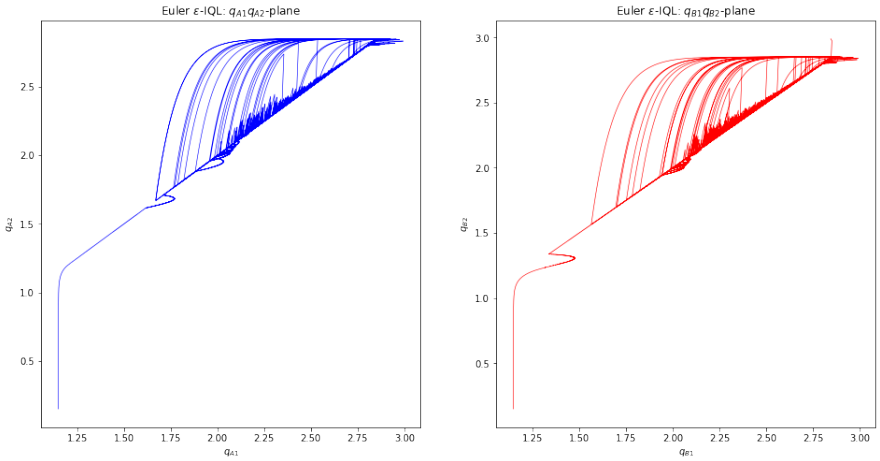


Figure 6: Plots showing the evolution of the Q-values in the $q_1q_2$-plane ($\epsilon = 0.011$, $h = 10^{-3}$, 1000 time units). Becomes transient and settles on continued mutual defection eventually.

(Figure 4), but for $h = 2 \times 10^{-3}$, it takes more than 200 time units (See Figure 8). For $h = 0.25 \times 10^{-3}$ (See Figure 9), the players do not seem to settle down at all (for at least more than 4000 time units). Thus Euler $\epsilon$-IQL depends sensitively on the timestep $h$ also, and we cannot expect some singular limit of Euler $\epsilon$-IQL when taking $h \to 0$.

We even have sensitive dependence on initial values; See Figure 10 where we ran Euler $\epsilon$-IQL for $Q_A = (2.7, 2.701)$, $(2.8, 2.801)$, $(2.9, 2.901)$.

A natural question to ask is then: Will players always eventually settle on continued mutual defection, or can we have non-transience (chaos)? For simulations I ran, most simulations settled down after 5000 time units or less, indicating that chaos is either not there or hard to find. However, a simulation with $\epsilon = 0.1$, $h = 10^{-3}$ and initial values $Q_A = (2.8, 2.801)$, $Q_B = (2.845, 2.99)$ refused to settle to mutual defection even after $10^4$ time units. It may be that there is a chaotic attractor, but with a very small basin of attraction. It is not clear what happens as $h \to 0$ either, as the behaviour seems to vary wildly as we decrease $h$, and running Euler $\epsilon$-IQL for very small $h$ is untenable for my machine.

## 4.2 Patterns of Prisoners

One note that we can make of all our simulations is that they seem to follow some repeating cycle. For $\epsilon = 0.1$, $h = 10^{-3}$ and initial values $Q_A = (2.8, 2.801)$, $Q_B = (2.845, 2.99)$, this cycle seems to last for 20 time units (See Figure 14). The cycle has 5 regimes [14]: I. Mutual Cooperation, II. Betrayal, III. Rapid Sliding, IV. Opportunistic Defection, and V. Stubborn Defection.

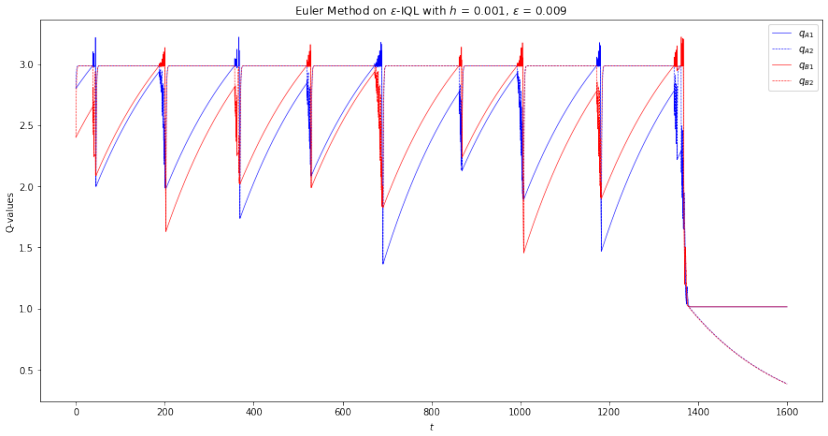


Figure 7: Transient behaviour after 1400 time units.

### 4.2.1 Mutual Cooperation

The longest regime, lasting about 12 time units. This occurs when both players are below the diagonal and cooperating repeatedly (following R4 dynamics seen in Figure 2). The motion in this phase is smooth as we are totally inside $R_4$. This motion continues until one player hits the diagonal and defects. From here we enter Betrayal.

### 4.2.2 Betrayal

Once a player hits the diagonal and defects once (in Figure 15, this is player B in red), the dynamics for the both players switch. The player who hit the diagonal follows the vector field in the right hand side of Figure 3 and is attracted towards $C$. Meanwhile the other (betrayed) player (in Figure 15, this is player A in blue) follows the vector field on the left hand side, experiencing a strong downward motion towards the diagonal and a sudden drop in the Q-value for cooperation. This regime last a mere 0.5 time units, until the other player (red) hits the diagonal and we enter the Rapid Sliding phase.

### 4.2.3 Rapid Sliding

This regime also lasts for only 0.5 time units. Once the betrayed player hits the diagonal, and we enter $R_1$ dynamics (right hand side of Figure 2), which pushes the betrayed player back over the diagonal. See Figure 15. The betrayed player (blue) is then pushed back over the diagonal, and continues in this fashion of rapidly jumping back and forth over the diagonal. While the second player was dropping towards the diagonal in the betrayal phase, the betraying player (red) moved towards $C$ away from the diagonal. In the Rapid Sliding phase, the betraying player's Q-value for defection (solid red) drops and moves towards the

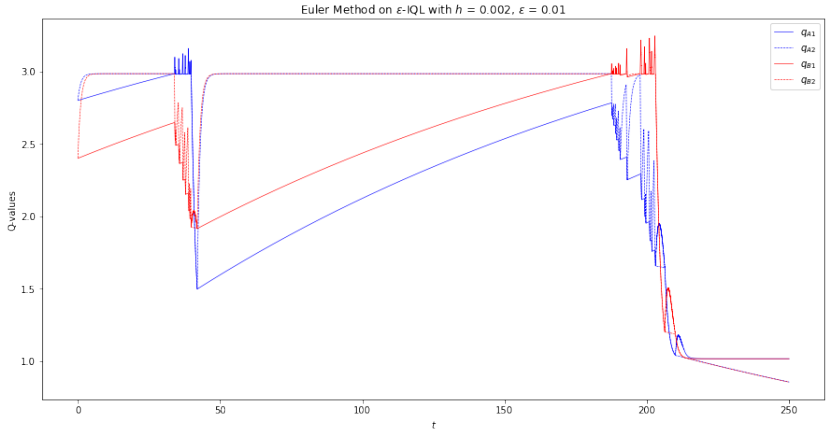


Figure 8: Transient behaviour after 200 time units.

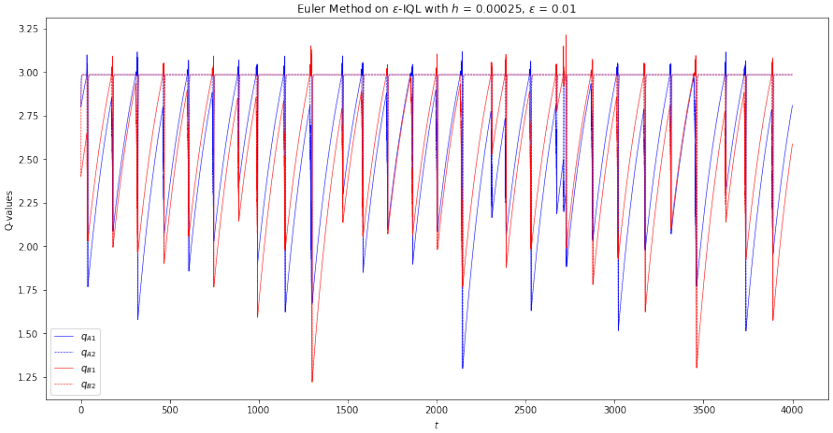


Figure 9: No apparent transient behaviour even after 4000 time units.

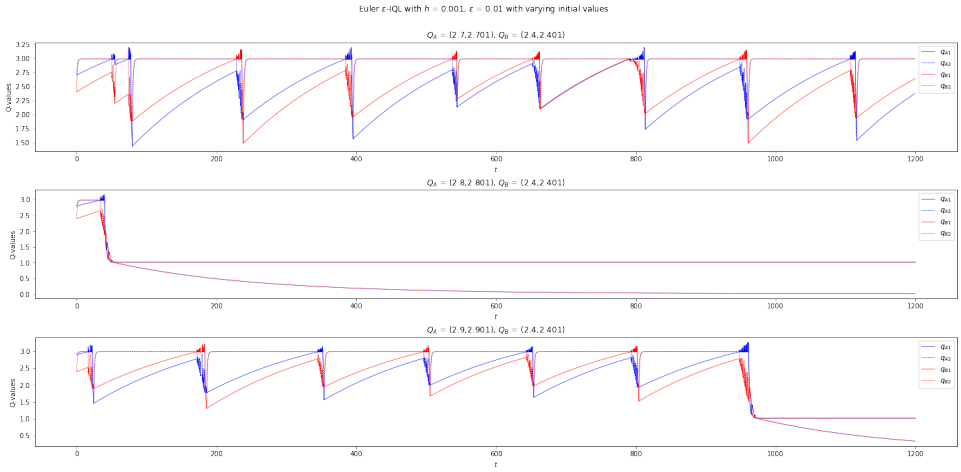


Figure 10: Transient behaviour for $Q_A = (2.8, 2.801)$ & $(2.9, 2.901)$, but not for $Q_A = (2.7, 2.701)$.

diagonal, away from $C$. When it crosses, we enter the Opportunistic Defection phase.

#### 4.2.4 Opportunistic Defection

This regime lasts around 6 time units. It is similar to the previous phase in which one player was rapidly jumping back and forth over the diagonal, except now both players are oscillating in this fashion (a kind of dual sliding). See Figure 16. The net motion for player A in blue is down the diagonal, while the Q-values for the opportunistically defecting player B in red barely moves. Note player B must stay above $C^\star$ in this phase otherwise it is pulled away from the diagonal and we enter another Mutual Cooperation phase. Eventually, player A stays below and away from the diagonal for long enough for player B to begin quickly sliding downwards, entering the Stubborn Defection phase. It is not clear what causes player B to exit this dual sliding phase. See Figure 17 to see the microscopic behaviour of this phase.

#### 4.2.5 Stubborn Defection

This regime lasts for around 1.25 time units. See Figure 18. Player A stays below the diagonal, stubbornly defecting. This causes the Q-value of defection for player B to drop rapidly until it hits the diagonal and stays on it, sliding downwards. Eventually player A experiences net motion towards the diagonal. When it hits, both players begin cooperating and we enter another Mutual Cooperation phase. Note that although the motion in this phase looks as smooth as the motion in Mutual Cooperation, a close up zoom shows that both players

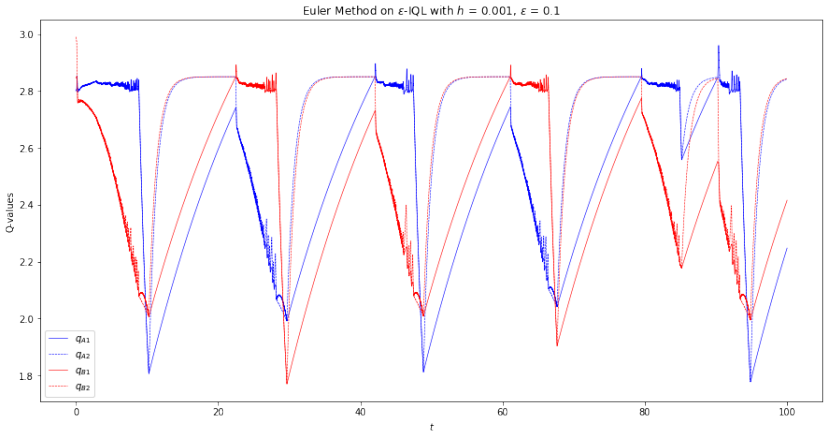


Figure 11: Chaotic Euler $\epsilon$-IQL ($\epsilon = 0.1$, $h = 10^{-3}$) early in its evolution.

are rapidly changing directions on a small scale, see Figure 19. See Proposition 4.1 for more discussion on this.

## 4.3 Sliding Dynamics

It seems as though the motion in the Opportunistic Defection phase is incredibly difficult to predict: In each time step both players can either jump over the diagonal and enter a new vector field, or one player can just fall short of the diagonal and both players follow a completely different vector field. Since the length of jumps depends on the timestep $h$, it is more clear how a slight variation can lead to completely different behaviour on the microscopic scale, which then has a knock-on macroscopic effect on the time to continued mutual defection.

Not all is lost if we attempt to find the net motion of the players under certain conditions over $N$ time steps. For this we will need to simplify our vector field somewhat by taking $\epsilon \to 0$, so that the motion in each time step is either purely vertical or purely horizontal. The following argument was introduced to me by my supervisor Professor Sebastian van Strien in [15].

**Proposition 4.1.** *Let player $A$ be very close but slightly below the diagonal i.e. $q_A = (q, q-\delta)$ for $0 < \delta \ll 1$. Let player $B$ be below and far from the diagonal i.e. $q_B = (p, r)$ with $r - p < 0$ and let $p < C_1 = 4 - \frac{3\epsilon}{2}$. As $\epsilon \to 0$, the motion of $q_A$ and $q_B$ as $h \to 0$ is given by:*

$$\dot{q} = -\frac{q(q-1)}{2q-1},\ \dot{p} = \frac{5q-4}{2q-1} - p,\ \dot{r} = 0 \tag{6}$$

Note that because the square with $C$ and $D$ as its edges is a trapping region for the flow, we do not have to worry about any $\dot{q}$ or $\dot{p}$ being undefined. Note also

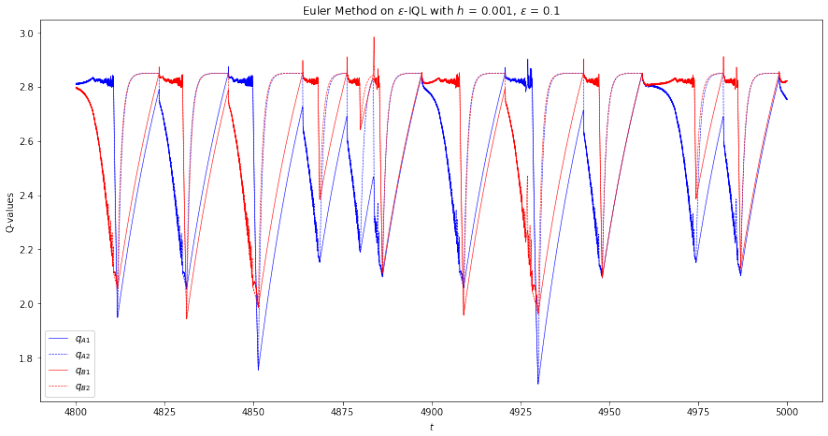


Figure 12: Chaotic Euler $\epsilon$-IQL ($\epsilon = 0.1$, $h = 10^{-3}$) much later in its evolution.

that this will describe the motion in both the Rapid Sliding and Opportunistic Defection (swapping $A$ and $B$ as needed).

*Proof.* We start with both players defecting, so we follow the vector field given in $R_1$. $A$ will quickly hit the diagonal, causing us to follow the vector field in $R_3$. This vector field pushes us back into $R_1$, so we will have rapid switching between these two. Taking $\epsilon \to 0$ (so that $\epsilon_0 \to 0, \epsilon_1 \to 1$), the equations of motion become:

$$\begin{aligned} R_1 &: \dot{q}_{A1} = 1 - q_{A1}, \quad \dot{q}_{A2} = 0, \quad \dot{q}_{B1} = 1 - q_{B1}, \quad \dot{q}_{B2} = 0 \\ R_3 &: \dot{q}_{A1} = 0, \quad \dot{q}_{A2} = -q_{A2}, \quad \dot{q}_{B1} = 4 - q_{B1}, \quad \dot{q}_{B2} = 0 \end{aligned}$$

So as $\epsilon \to 0$, the players become less indecisive and the vector field becomes more unidirectional; $B$ moves only horizontally as $q_{B2}$ is a constant in both vector fields, while $A$ is switching between vertical downward and horizontal leftward motion. In what follows we write $dt$ as the time step instead of $h$.

Let us focus on player $A$ first. We have its velocity in both directions, and we will use this to determine how much time it spends above and below the diagonal. Let $N$ be the total number of time steps in some time range. Say that $L$ of these are leftward steps from under the diagonal and $D$ of them are downward steps from above the diagonal. We must have $L + D = N$. The total downward displacement is then given by $(-q)Ddt$ and the total leftward displacement given by $(1-q)Ldt$. If $A$ stays very close to the diagonal (since it is being pushed onto it from both directions), we must have that $(-q)Ddt = (1-q)Ldt$. Combining this with $L + D = N$ gives

$$L = \frac{Nq}{q-1} \text{ and } D = \frac{N(q-1)}{2q-1},$$

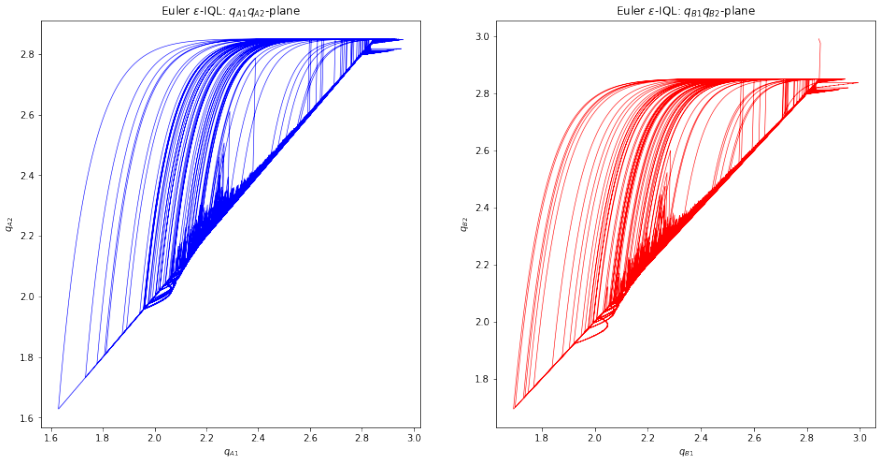


Figure 13: Plots showing the evolution of the Q-values in the $q_1q_2$-plane ($\epsilon = 0.1$, $h = 10^{-3}$, 1000 time units). It certainly looks chaotic.

meaning that the total displacement of $q$ after $N$ steps is on average equal to

$$Ndq = (-q)Ddt = (1-q)Ldt \implies \frac{dq}{dt} = -\frac{q(q-1)}{2q-1}.$$

Now for player $B$. It will spend the same number of time steps under the influence of vector fields $R_1$ and $R_3$ as $A$ does. Firstly, $\dot{q}_{A2} = \dot{r} = 0$ as $B$ experiences no vertical motion in either vector field. Secondly, using expressions for $L$ and $D$ from before,

$$dp = (4-p)\frac{D}{N}dt + (1-p)\frac{L}{N}dt = dt\left(\frac{5q-4}{2q-1} - p\right)$$

i.e. $\frac{dp}{dt} = \frac{5q-4}{2q-1} - p = \dot{p}$ in the limit $dt \to 0$. □

This seems to agree with our Euler $\epsilon$-IQL simulations; See Figures 20 and 21 to see how well they seem to agree. That is, up until the point where $B$ hits the diagonal also, where the $\epsilon$-IQL system will change vector fields but the smoothed version does not.

Thus it seems it is possible to "fill in the gaps" and obtain a vector field in the undefined region just by looking at the vector fields around it, at least in some places like $\{D_A = 0, D_B \neq 0\}$. We did make some heavy assumptions in the above proof: we assumed that player A would stay on the diagonal, and that player B does not cross the diagonal at any point. We can also note that $1 - q \leq \dot{q} = -\frac{q(q-1)}{2q-1} \leq 0$, where the bounds are the two velocities that player A experiences, so we have not "added" any velocities that were not already there. The same can be said of $\dot{p}$.

We shall now depart from our study of $\epsilon$-IQL in particular and find a general

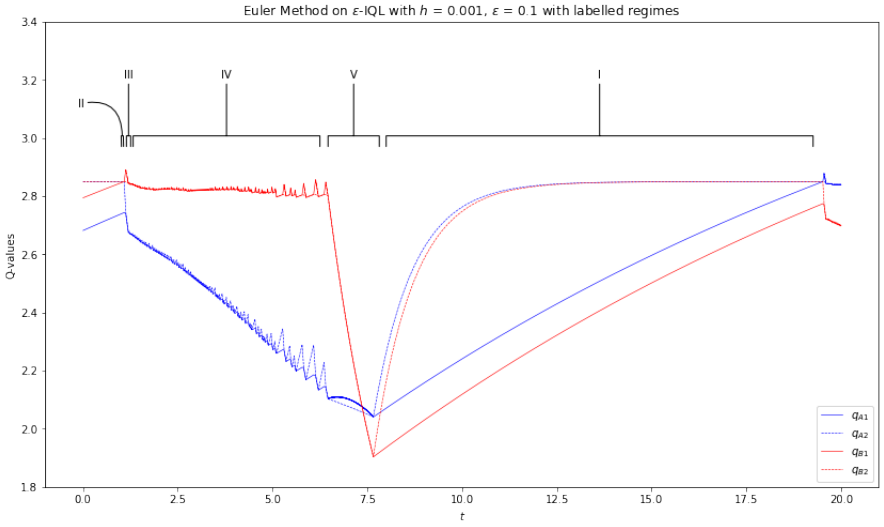


Figure 14: A full cycle of a typical Euler $\epsilon$-IQL evolution, with labelled regimes.

theory to place it in. One well-developed area of study is the study of differential equations with discontinuous right-hand sides and Filippov theory, which sees just how far we can get when applying these heavy assumptions.

# 5 Piecewise Smooth Dynamics

## 5.1 Vector Fields with Gaps

Dynamical systems is a well studied field, producing many powerful tools to study and understand a wide range of problems, from symbolic dynamics detecting chaos in circle mappings, to the Poincaré-Bendixson theorem classifying all orbits of ODEs in the plane. These have applications in computer simulation [16], fluid flows, optical systems and even game theory. However, many significant practical problems are excluded by the requirement that the evolution of the system is defined by a smooth function of its arguments. Thus, much of the qualitative behaviour of systems that we may sometimes call pathological remains unstudied. Built-in discontinuities occur in the study of dry friction, genetics [17] and electromechanics [18]. For an example of a discontinuous mechanical system, we can consider a model of a mass with a dry friction damper as seen in Figure 22 [19]. Here the equations of motion will be given by:

$$m\ddot{x} + kx = f_0 \cos \omega t - f(\dot{x}) \text{ with } f(\dot{x}) = \begin{cases} -\mu, & \text{if } \dot{x} < 0 \\ +\mu, & \text{if } \dot{x} > 0 \end{cases}$$

In any model of this system, we will have the friction opposing the direction of motion. However, it is not clear what the force of the friction should be

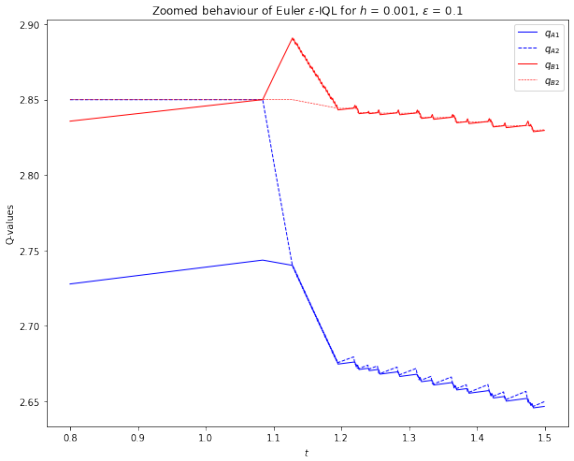


Figure 15: Player B in red hitting the diagonal, causing a Betrayal ($t \in (1.1, 1.15)$) followed by a Rapid Sliding Phase ($t \in (1.15, 1.2)$).

when $\dot{x} = 0$ when the mass is motionless. arctan functions are usually used to continuously interpolate between $\mu$ and $-\mu$ so that the friction force is given as a single valued function of the velocity that is defined at $\dot{x} = 0$. From here, standard ODE techniques can be applied. However, in practice, this leads to numerical difficulties in integration due to the steepness of the interpolation and crucially does not properly describe stiction, a tribilogical phenomenon arising from dry friction whereby the force needed to overcome friction when the body is motionless is greater than when it is moving [20]. Thus, a set-valued extension of the frictional force[2] in the model allows for the friction force to take a range of values at $\dot{x} = 0$ i.e. we set

$$f(\dot{x}) \in F(x) = \begin{cases} -\mu, & \text{if } \dot{x} < 0 \\ [-\mu, \mu], & \text{if } \dot{x} = 0 \\ +\mu, & \text{if } \dot{x} > 0 \end{cases}$$

By embedding our discontinuous $f(x)$ into a set valued map, we have a more useful model [21], as a differential inclusion rather than a differential equation. The set-valued extension $F$ is also subject to techniques from the study of differential inclusions, which guarantees existence of solutions under certain conditions [19]. The conditions described below allow $F$ enough regularity to have trajectories closed related to the trajectories of the original differential equation.

**Definition 5.1** (Upper Semi-Continuity)**.** A set valued function $F(x)$ is upper semi-continuous in $x$ if for $y \to x$

$$\sup_{a \in F(y)} \inf_{b \in F(x)} \|a - b\| \to 0$$

[2]This extension still excludes stiction phenomena, but is convex and upper semi-continuous so that we will have solutions and is sufficient for our discussion.

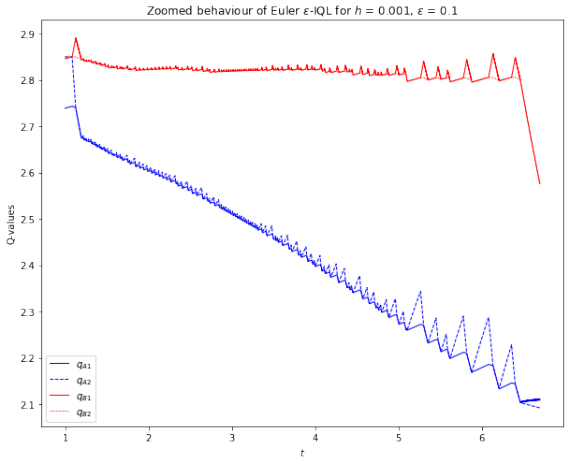


Figure 16: Player B in red opportunistically defecting, causing Q-values for player A in blue to slide downwards.

**Definition 5.2** (Closed Convex Hull)**.** For a set $A \subset \mathbb{R}^n$, $\overline{\text{co}}\, A$ denotes the closed convex hull of $A$, i.e.

$$\overline{\text{co}}\, A = \left\{ \sum_{i=1}^{n+1} \lambda_i x_i : x_i \in \bar{A}, 0 \leq \lambda_i \leq 1, i = 1, \ldots, n+1, \sum_{i=1}^{n+1} \lambda_i = 1 \right\}$$

**Definition 5.3.** A set $A$ is said to be closed convex if $\overline{\text{co}}\, A = A$. A set-valued map $F : \mathbb{R}^n \to \mathbb{R}^n$ is said to be closed and convex if $\forall x \in \mathbb{R}^n$, $F(x)$ is closed convex.

**Definition 5.4** (Absolute Continuity)**.** A function $x : [a, b] \to \mathbb{R}^n$ is absolutely continuous if $\forall \epsilon > 0$, $\exists \delta$ such that, for any countable collection of disjoint subintervals $[a_k, b_k]$ of $[a, b]$ such that

$$\sum_k (b_k - a_k) < \delta,$$

we have

$$\sum_k \big| x(b_k) - x(a_k) \big| < \epsilon.$$

An absolutely continuous function is both continuous and of bounded variation, and so have a finite derivative except at most on a set of measure zero. The following was then proved by Aubin and Cellina [22].

**Theorem 5.5.** *Let $F$ be a set-valued function that is upper semi-continuous, closed, convex and bounded for all $x \in \mathbb{R}^n$. Then, for each $x_0 \in \mathbb{R}^n$, there exists*

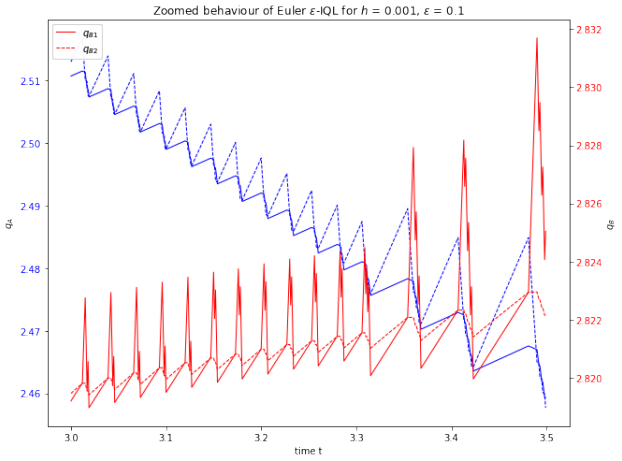


Figure 17: Close up of the Opportunistic Defection phase.

*$\tau > 0$ and an absolutely continuous function $x(t)$ defined on $[0, \tau]$ which is a solution of the initial value problem*

$$\dot{x} \in F(t, x(t)),\ x(0) = x_0.$$

Thus, the set-valued extension of our dry friction damper model has a continuous solution, albeit not a unique one. To review, we had a situation in which we knew some values for our frictional force (negative to the direction of travel) and were able to "fill in the gaps" for $\dot{x} = 0$ with a set-valued function, from which we were able to retrieve a solution to the IVP. We shall aim to continue this, and figure out where the Euler $\epsilon$-IQL trajectories come into all of this.
Our our object of study will be piecewise smooth dynamical systems (PWS): IVPs in which the right hand side varies discontinuously (or is not defined) as the solution trajectory approaches one or many separating surfaces[3], but is smooth everywhere else.

**Definition 5.6** (Piecewise Smooth System (PWS))**.** A piecewise smooth system is given by a finite set of ODEs

$$\dot{x} = f(x),\ f(x) = f_i(x) \text{ for } x \in R_i,\ i = 1, \ldots, m$$

where $\cup_i R_i = \mathcal{D} \subset \mathbb{R}^n$, each $R_i$ has non-empty interior, $\bar{\mathcal{D}} = \mathbb{R}^n$ and $\mathbb{R}^n \backslash \mathcal{D}$ has zero Lebesgue measure. For each $i, j$, the intersection $\bar{R_i} \cap \bar{R_j}$, known as a separating surface, is either an $\mathbb{R}^{n-1}$ dimensional manifold included in the boundaries $\partial R_i$ and $\partial R_j$, or is the empty set. Each vector field $f_i$ is smooth in $x$, and defines a smooth flow $\Phi_i(x, t)$ within any open set $U \subset R_i$.

The values at which a solution trajectory reaches a separating surface are called events, and we shall henceforth assume our events are isolated i.e. we

[3] Also known as discontinuity or switching surfaces

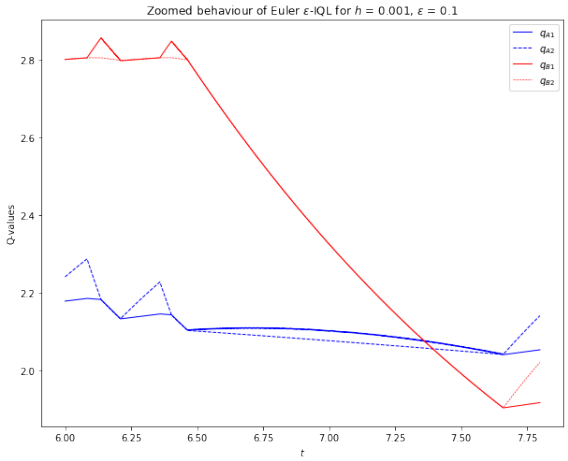


Figure 18: Player A in blue stubbornly defecting, causing Q-values for player B in red to slide downwards.

cannot have an infinite number of crossings of these surfaces in a finite time.

Note that in Definition 5.6, we have no rule for how the dynamics should evolve within any of the $\bar{R}_i \cap \bar{R}_j$. One possibility is to arbitrarily place $\bar{R}_i \cap \bar{R}_j$ within $R_i$, so that we only follow $f_i$ there. We shall see that such arbitrary rules make little difference to the dynamics unless the flow becomes confined to $\bar{R}_i \cap \bar{R}_j$, which are known as Filippov sliding trajectories.

We will assume that the separating surfaces are non-empty and are implicitly defined smooth surfaces of co-dimension 1 i.e. $\forall i, j \, \exists$ differentiable $h_k : \mathbb{R}^n \to \mathbb{R}$ such that $\bar{R}_i \cap \bar{R}_j = h_k^{-1}(0) \coloneqq \Sigma_k$, $k = 1, \ldots, p$. The intersection of every separating surface, known as *the* discontinuity surface, is then defined by

$$\Sigma \coloneqq \{x \in \mathbb{R}^n : h(x) \coloneqq (h_1(x), \ldots, h_p(x)) = \mathbf{0}\}$$

If we require that $\{\nabla h_k(x)\}_{k=1,\ldots,p}$ forms a basis for $\mathbb{R}^p$, then $\Sigma = h^{-1}(0)$ is a codimension $p$ (dimension $n-p$) surface. Furthermore, we will assume that each $R_i$ is defined by some combination of the $h_k$s being greater than or less than zero i.e. $m = 2^p$. See Figures (23) and (24) for examples. As we will see, prescribing a vector field on the discontinuity surface $\Sigma$ when $p \geq 2$ is difficult, but as in Proposition 4.1 we can choose a reasonable vector field when using the Euler method. We start with a simple example.

As we saw with the Proposition 4.1, it is possible to prescribe a reasonable solution in some regions for a differential equation that is not well-defined everywhere using the Euler method. We shall start with a simple example.

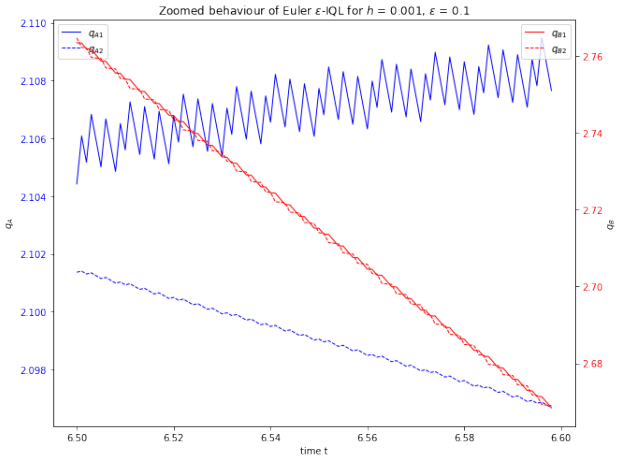


Figure 19: Close up of the Stubborn Defection phase.

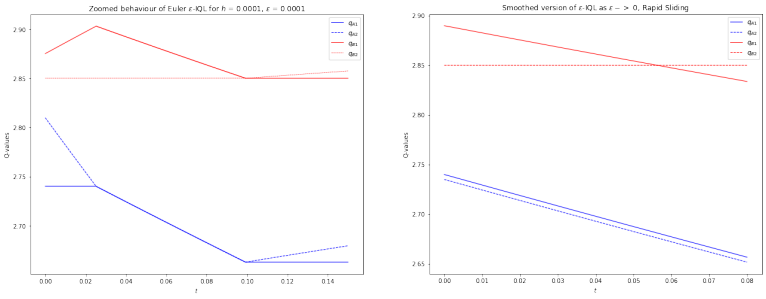

Figure 20: On the left: A Rapid Sliding phase of Euler $\epsilon$-IQLGraph with $\epsilon = 0.0001$. On the right: Graph of the trajectories described in (6). The initial value is the point where the graph on the right first hits the diagonal.

**Example 5.7.** We consider the codimension $p = 1$ PWS

$$\dot{x} = f(x),\ x = (x_1, x_2) \in \mathbb{R}^2, f(x) := \begin{cases} f_+(x) = (+1, -1), & \text{if } x_2 > 0 \\ f_-(x) = (+1, +1), & \text{if } x_2 < 0 \end{cases} \tag{7}$$

Our separating/discontinuity surface is given by $\Sigma = \Sigma_1 = \{(x_1, x_2) : x_2 = 0\} = h_1^{-1}(0)$, where $h_1(x_1, x_2) = x_2$. Define $R_- = \{h_1 < 0\}$ and $R_+ = \{h_1 > 0\}$.

Our vector field is not defined on $\Sigma$. How do we go about filling in this gap? We could replace $f$ by a set-valued function $F$, where $F(x) = f(x)$ for $x_2 \neq 0$ and $F(0) = \{(1, a) : a \in [-1, +1]\}$, in a similar manner to the friction model from before. However, if we try to apply Euler method, a more "natural" extension of $f$ presents itself. Let us start by considering the Euler method with timestep $dt$ applied to this system.

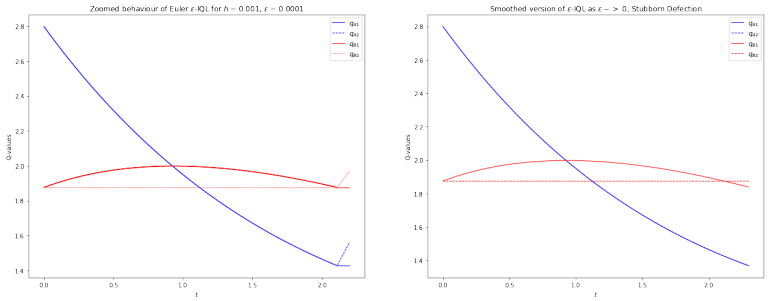


Figure 21: On the left: A Stubborn Defection phase of Euler $\epsilon$-IQLGraph with $\epsilon = 0.0001$. On the right: Graph of the trajectories described in (6). The initial value is the point where the graph on the right first hits the diagonal.

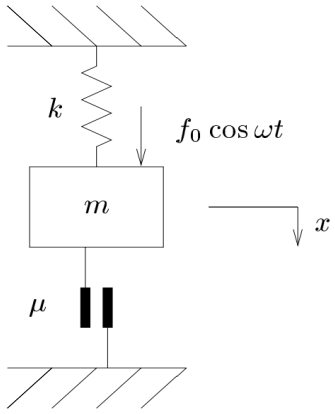


Figure 22: A system in which the dry friction force is a discontinuous function of velocity, taken from [19].

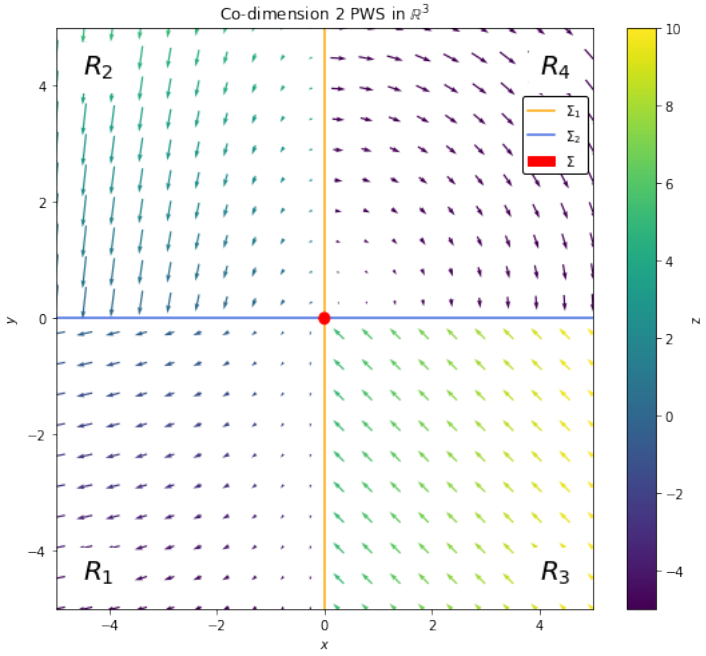


Figure 23: An example of a codimension $p = 2$ PWS with $h_1(x, y) = x$ and $h_2(x, y) = y$. Labelling is as in (11). The color represents the component in the $z$ direction.

For $x$ such that $|x_2| > dt$, $f(x)$ is smooth and so the Euler iterates will approach a line as $dt \to 0$ in this range.
For $|x_2| < dt$, we have that $x(t+dt)$ will land on the opposite side of separating line $\Sigma$. Of course, once it jumps from below to above $\Sigma$, it will jump from above to below in the next time step (and vice versa). Let $N$ be the total number of time steps, with $U$ of these being from below to above and $D$ vice versa. Then $U + D = N$, and the total $x_2$ displacement will be given by $Ndx_2 = U(+dx_2) + D(-dx_2)$. Since every jump downwards is immediately followed by a jump upwards, we must have $U = D = \frac{N}{2}$, giving the displacement as $dx_2 = (\frac{N}{2} - \frac{N}{2})dt = 0 \implies \frac{dx_2}{dt} = 0$. The total $x_1$ displacement will be given by $Ndx_1 = U(+dt) + D(+dt) = Ndt \implies \frac{dx_1}{dt} = +1$.
For $|x_2| = dt$, we can slightly decrease $dt$ so that $x(t + dt)$ is on the same side as $x(t)$ and such that $x(t + 2dt)$ is on the other side i.e. we can assume that $x$ always "jumps" to the other side.
Thus, for $dt \to 0$, we can define $\dot{x}$ for all $x \in \mathbb{R}^2$ as:

$$\dot{x} = F_s(x) = \begin{cases} (+1, -1), & \text{if } x_2 > 0 \\ (+1, 0), & \text{if } x_2 = 0 \\ (+1, +1), & \text{if } x_2 < 0 \end{cases}$$

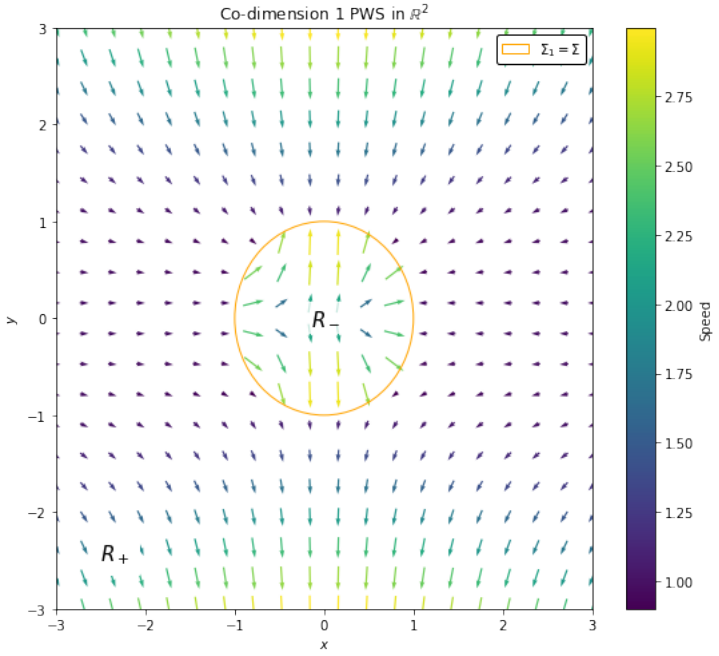


Figure 24: An example of a codimension $p = 1$ PWS with $h_1(x, y) = x^2 + y^2 - 1$. The color represents the size of the vectors.

Although the $F_s(x)$ is not continuous, $x(t)$ is absolutely continuous (differentiable everywhere except when it first hits $\Sigma$), and is given by:

$$x(t) = \begin{cases} (x_1(0) + t, x_2(0) - t \operatorname{sgn}(x_2(0))), & \text{for } t \in [0, |x_2(0)|] \\ (x_1(0) + t, 0), & \text{for } t \in [|x_2(0)|, +\infty] \end{cases}$$

There were of course, many other methods that would have given a singularly-valued RHS to problem (7), as discussed in Section 5.3.

We can note that $F_s(x) \in \overline{\text{co}}\{f_-(x), f_+(x)\}$. Intuitively, we should require that any new extended vector field should be somewhere in between the surrounding vector fields; we cannot "add" any more speed. This motivates the definition of a Filippov vector field [23].

**Definition 5.8** (Filippov Vector Field)**.** For a PWS,

$$\dot{x} = f(x), f(x) = f_i(x) \text{ for } x \in R_i, i = 1, \ldots, m$$

we define a new multivalued function $F$, known as the Filippov solutions or Filippov vector field, as

$$F(x) := \bigcap_{r>0} \bigcap_{A \in \mathcal{N}} \overline{\text{co}}\{f(y) : |y - x| < r, y \notin A\}$$

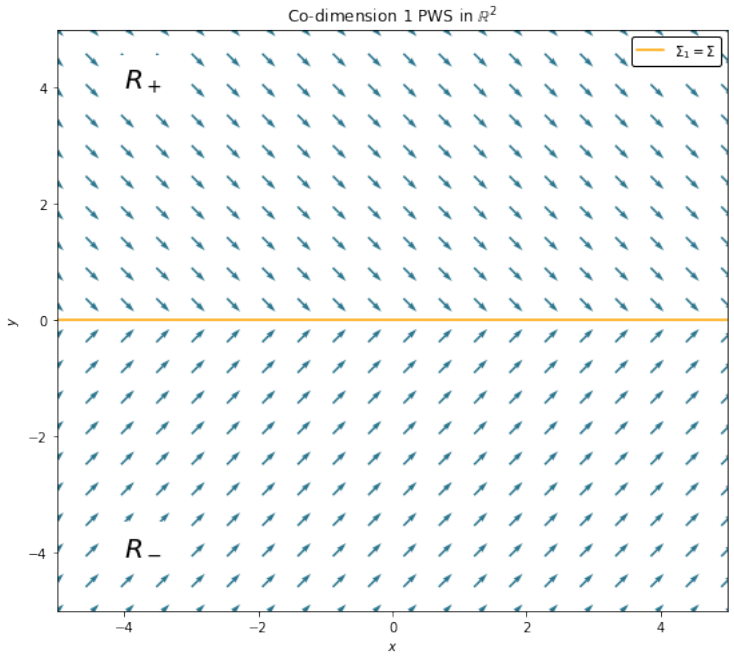


Figure 25: Plot of the PWS given in (7) with $h_1(x, y) = y$.

where $\mathcal{N} = \{A \subset \mathbb{R}^n : \lambda(A) = 0\}$, where $\lambda$ denotes the Lebesgue measure on $\mathbb{R}^n$. In other words, $F(x)$ is the closed convex hull of the set of all limits of $f(y)$ as $y \to x$.

Note that for $x \in R_i$, $F(x) = \{f_i(x)\}$. If $x$ is in the separating surface between $R_i$ and $R_j$, but not included in any other separating surface,

$$F(x) = \overline{\text{co}}\{f_i(x), f_j(x)\} = \{\alpha f_i(x) + (1-\alpha) f_j(x) : \alpha \in [0,1]\}.$$

Note also that even if $f(x)$ was defined on $R_i \cap R_j$, the values of $f$ are thrown away here, since $R_i \cap R_j$ is of measure zero[4].

Since every open neighbourhood of $\Sigma$ has non-empty intersection with each $R_i$, if $x \in \Sigma$ then

$$F(x) = \overline{\text{co}}\{f_i(x)\}_{i=1,\dots,2^p} = \left\{ \sum_{i=1}^{2^p} \lambda_i f_i(x) : \lambda_i \in [0,1], \sum_{i=1}^{2^p} \lambda_i = 1 \right\}$$

We shall omit the dependence of the $\lambda_i$s on $x$ for specific elements of $F(x)$ where it is obvious.

[4]This is in contrast to the larger set of Krasovskii solutions, which does consider measure zero sets.

The set of Filippov solutions is a closed, convex and upper semi-continuous set, and hence we can apply Theorem 5.5 to reason that extending our PWS to the differential inclusion

$$\dot{x} \in F(x)$$

will give us absolutely continuous trajectories. This at least meets our criteria of existence of solution trajectories, but these will not be unique. However, we saw in Example 5.7 that when $\Sigma$ is attractive we could have a single Filippov solution $F_s(x)$ that would give a unique trajectory that was confined to $\Sigma$. $F_s(x)$ is called a Filippov sliding solution and they can be guaranteed to exist given some conditions on the vector fields in a neighbourhood of $\Sigma$.

## 5.2 Filippov Systems and Sliding

There are in essence 3 possible different situations for what the PWS vector field looks in a neighbourhood any of the separating surfaces $\Sigma_k$:

1. Transversal Crossing: $x$ is attracted to $\Sigma_k$ on one side and repelled on the other.
2. Attractive Sliding: $x$ is attracted to $\Sigma_k$ from both sides.
3. Repulsive Sliding: $x$ is repelled from $\Sigma_k$ from both sides.

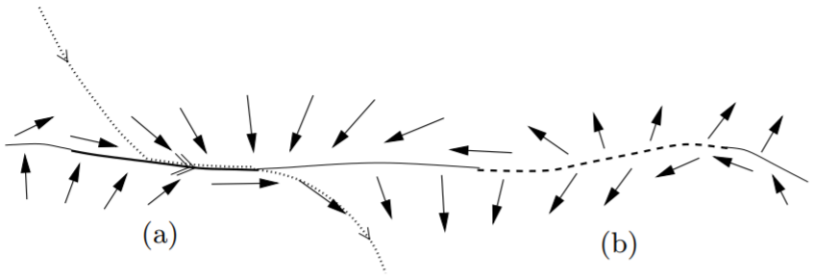


Figure 26: A typical separating surface of a two-dimensional Filippov system showing the behavior of the vector fields on both sides. Bold and dashed regions represent where (a) attracting and (b) repelling sliding occurs, respectively. All other regions are where transveral crossing occurs. Dotted lines indicate three individual trajectory segments. Taken from [24].

If the vector field is transverse near $\Sigma_k$, then it does not in practice matter what vector field we choose on $\Sigma_k$: since every element of $F(x)$ is also pointing out of $\Sigma_k$ for $x \in \Sigma_k$, a trajectory $x(t)$ will only be in $\Sigma_k$ for a single time instant. If we interpret a solution to our PWS instead as the solution to the integral inclusion[5]

$$x(t) \in x(0) + \int_0^t F(x)dt$$

[5]The very weak Carathéodory solution concept.

then the part where $F(x)$ is multivalued i.e. when $x \in \Sigma_k$ will have measure zero, and so any Filippov vector field will give the same trajectory.

If the vector field is repulsive sliding near $\Sigma_k$, no trajectory can approach $\Sigma_k$ in forward time and so any Filippov vector field there cannot have any effect on any trajectories.

The only interesting case is when the vector field is attractive near $\Sigma_k$. At least one of the Filippov vector fields in $F$ will be parallel to $\Sigma_k$, causing the trajectory to stay within $\Sigma_k$. As we saw in Subsection , we can have one player sliding (the vector fields are attractive sliding near one of $\{D_A = 0\}$ or $\{D_B = 0\}$ as in Rapid Sliding or Stubborn Defection), or both players sliding at once (the vector fields are attractive sliding near $\{D_A = D_B = 0\}$ as in Opportunistic Defection). In Example 5.7, we saw that the Euler iterates would "stick" to $\Sigma = \{x_2 = 0\}$ and slide along it. We shall see what sliding looks like for different values of the codimension $p$.

### 5.2.1 Codimension $p = 1$ case

Let us consider the general codimension 1 PWS, given by

$$\dot{x} = f(x), f(x) = \begin{cases} f_+(x), & \text{for } h(x) > 0 \\ f_-(x), & \text{for } h(x) < 0 \end{cases} \tag{8}$$

Our regions are $R_+ = \{x : h(x) > 0\}$ and $R_- = \{x : h(x) < 0\}$ and our single separating surface is $\Sigma_1 = \Sigma = \{x : h(x) = 0\}$. Any Filippov vector field on $\Sigma$ would be given by

$$F(x) = \alpha f_+(x) + (1 - \alpha) f_-(x),\ \alpha \in [0, 1]$$

i.e. a convex combination of $f_+$ and $f_-$.

One possible interpretation of the Filippov vector field in this case can be through "fuzzy logic" [28]. The "fuzzification" of a logical statement such as "$h(x)$ is positive" is given by a statement $\alpha : \mathbb{R} \to [0, 1]$ (instead of the usual $\{0, 1\}$) such that $\alpha = 0$ when the logical statement is definitely false, when $h(x)$ is definitely or "very" negative and $\alpha = 1$ for when $h(x)$ is "very" positive. Intermediate values of partial truth have $\alpha \in (0, 1)$, when $h(x)$ is close to zero. This allows us to interpret our Filippov field as a "fuzzification" of (8), an intuition we will try to extend in Subsection 5.4.

**Definition 5.9** (Sliding Region)**.** The sliding region $\widehat{\Sigma}$ of a PWS of the form (8) is the parts of $\Sigma$ such that

$$(\nabla h(x)^T f_+(x)) \cdot (\nabla h(x)^T f_-(x)) < 0$$

i.e. $\nabla h(x)^T f_+(x)$ (the component of $f_+$ normal to $\Sigma$) and $\nabla h(x)^T f_-(x)$ (the component of $f_-$ normal to $\Sigma$) have opposite signs.

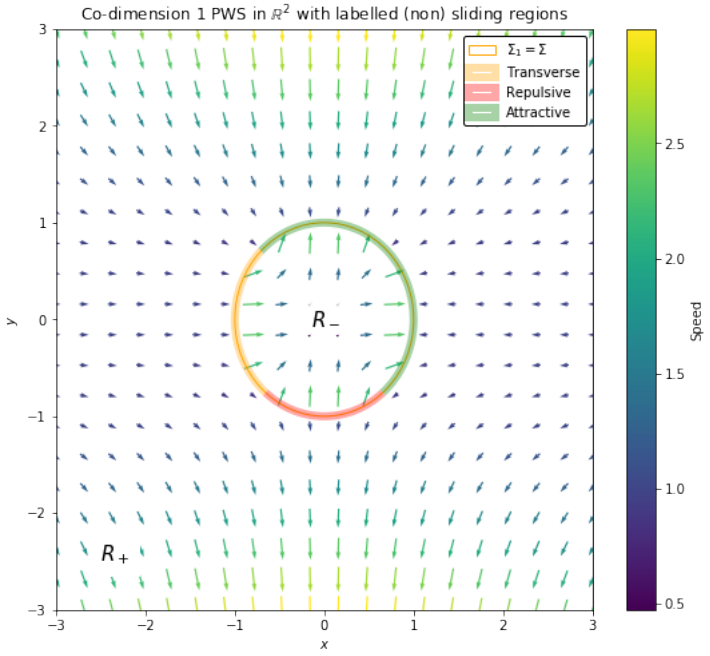


Figure 27: Plot of system (9) with transverse, repulsive sliding and attractive sliding portions shaded.

**Example 5.10.**

$$\dot{x} = f(x),\ x = (x_1, x_2) \in \mathbb{R}^2, f(x) := \begin{cases} f_+(x) = \frac{1}{\sqrt{x_1^2+x_2^2}}\left(-x_1, -x_2^2\right), & \text{if } x_1^2 + x_2^2 > 1 \\ f_-(x) = \frac{1}{\sqrt{x_1^2+x_2^2}}\left(3x_1^2, 3x_2^2\right), & \text{if } x_1^2 + x_2^2 < 1 \end{cases} \tag{9}$$

Here $h(x) = x_1^2 + x_2^2 - 1$, so $\nabla h(x) = (2x_1, 2x_2)$ which is surjective on $h^{-1}(0)$. The sliding regions can be found by finding where $(x_1^3 + x_2^3)(x_1^2 + x_2^3) > 0$, and is shown in Figure 27.

**Definition 5.11** (Filippov Sliding Vector Field)**.** For $x \in \widehat{\Sigma}$, the Filippov sliding vector field $F_s$ is an element of $F$ such that

$$\nabla h(x)^T F_s(x) = 0$$

i.e. a Filippov solution that is parallel to $\Sigma$.

**Lemma 5.1.** *$F_s$ is unique in the $p = 1$ case, given by*

$$F_s(x) = \frac{(\nabla h(x)^T f_+(x))f_+ - (\nabla h(x)^T f_-)f_-}{\nabla h(x)^T f_+(x) - \nabla h(x)^T f_-(x)}.$$

*Proof.* Any element of $F(x)$ can be written as $\alpha f_+(x) + (1-\alpha)f_-(x)$ for some $\alpha \in [0, 1]$. Therefore $\nabla h(x)^T F_s(x) = (1-\alpha)\nabla h(x)^T f_+(x) + \alpha \nabla h(x)^T f_-(x) =$

0. Since $x$ is assumed to be in the sliding region of $\Sigma$, $\nabla h(x)^T f_+(x)$ and $\nabla h(x)^T f_-(x)$ have opposite signs and therefore $\nabla h(x)^T f_+(x) - \nabla h(x)^T f_-(x) \neq 0$. Thus $\alpha = \frac{\nabla h(x)^T f_+(x)}{\nabla h(x)^T f_+(x) - \nabla h(x)^T f_-(x)}$ and $F_s$ is unique. □

Although this holds for both attractive and repelling sliding, the latter is usually considered ill-posed. As noted above, repelling sliding cannot have any effect on trajectories and we reason that we would be able to leave $\Sigma$ via $f_+$ or $f_-$ at any point if we found ourselves in a repulsive sliding region of $\Sigma$. Henceforth, when we talk of Filippov sliding, we consider only the case of attractive sliding and of $\widehat{\Sigma}$ being attractive. We will also stipulate that $\widehat{\Sigma}$ is attractive in finite time from now on. Next, we introduce some useful notation.

**Definition 5.12** (Normal Components)**.** For a PWS

$$\dot{x} = f(x),\ f(x) = f_i(x) \text{ for } x \in R_i,\ i = 1, \ldots, 2^p$$

we define the normal components of $f_i$ relative to $\Sigma_k$ as

$$w_i^k(x) = \nabla h_k^T f_i(x),\ k = 1, \ldots, p$$

These are how much each $f_i$ points towards $\Sigma_k$.

We shall omit the dependence of $w_i^k$ on $x$ where it is obvious.
For $p = 1$, we note that if, near $\Sigma_1$, we have:

1. $w_1^1 > 0, w_2^1 > 0$: $f_1$ is pointing away from $\Sigma$, $f_2$ is pointing towards $\Sigma$. This is transverse crossing from $R_2$ to $R_1$.
2. $w_1^1 < 0$, $w_2^1 < 0$: $f_1$ is pointing towards $\Sigma$, $f_2$ is pointing away from $\Sigma$. This is transverse crossing from $R_1$ to $R_2$.
3. $w_1^1 > 0$, $w_2^1 < 0$: Both $f_1$ and $f_2$ are pointing away from $\Sigma$. This is repulsive sliding.
4. $w_1^1 < 0$, $w_2^1 > 0$: Both $f_1$ and $f_2$ are pointing towards $\Sigma$. This is attractive sliding.

Now we can write the Filippov sliding solution from Lemma 5.1 more concisely as

$$F_s(x) = \frac{w_+^1 f_+(x) - w_-^1 f_-(x)}{w_+^1 - w_-^1}.$$

How does Filippov sliding relate to the Euler method? As we saw in Example 5.7, the Euler iterates converged to the Filippov sliding trajectory. Indeed, this holds in general:

**Proposition 5.2.** *Let $x_s(t)$ be a Filippov sliding trajectory of (8) i.e.*

$$\dot{x}_s(t) = F_s(x(t)) \text{ for } t \in [0, \tau]$$

*for some $\tau > 0$ and $x_s(0) \in \widehat{\Sigma}$. Let $\eta^h$ be the Euler polygon generated by the Euler method on (8) with time step $h$ and initial condition $x_0 \in \mathcal{N}_h$, where is $\mathcal{N}_h$ is an $h$-neighbourhood of $\widehat{\Sigma}$ with $\Sigma$ removed. Then $\eta^h(t)$ remains in an $h$-neighbourhood of $\Sigma$ for $t \in [0, \tau]$ i.e. as long as it remains attractive. Furthermore,*

$$\eta^h(t) \to x_s(t) \text{ uniformly as } h \to 0$$

*i.e. Euler iterates converge to sliding trajectories when $\Sigma$ is attractive.*

For a proof see [26] Chapter 3, section 1.1.

This is actually a particular case of Theorem 5.14. The reason that Euler method "realises" a unique trajectory is because the Filippov sliding solution is unique for $p = 1$. We can see that any Filippov sliding vector field $F_s(x) = \sum_{i=1}^{2} \lambda_i f_i(x)$ will need to satisfy:

$$\begin{pmatrix} 1 & 1 \\ w_1^1 & w_2^1 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

which has solutions for $w_1^1 \neq w_2^1$, and has solutions with $\lambda_i \in [0, 1]$ for $w_1^1$ and $w_2^1$ having different signs. So we can see that in the codimension $p = 1$ case, we can always prescribe a vector field onto the undefined area $\Sigma$ that will fully describe our trajectories that start outside $\Sigma$: On $\widehat{\Sigma}$ we will use the Filippov sliding vector field $F_s(x)$, and outside $\widehat{\Sigma}$, either no trajectories can approach $\Sigma$ as it is repulsive or trajectories will transversely cross $\Sigma$. Indeed we shall see that the codimension $p = 1$ case is the only such case in which the basic Filippov conditions prescribe a unique sliding vector field.

### 5.2.2 Codimension $p = 2$ case

We consider the system

$$\dot{x} = f(x), f(x) = \begin{cases} f_1(x), & \text{for } h_1(x) < 0,\ h_2(x) < 0 \\ f_2(x), & \text{for } h_1(x) < 0,\ h_2(x) > 0 \\ f_3(x), & \text{for } h_1(x) > 0,\ h_2(x) < 0 \\ f_4(x), & \text{for } h_1(x) > 0,\ h_2(x) > 0 \end{cases} \tag{10}$$

We shall label the regions as (WLOG):

$$\begin{aligned} R_2 &= \{h_1(x) < 0, h_2(x) > 0\} \quad & R_4 &= \{h_1(x) > 0, h_2(x) > 0\} \\ R_1 &= \{h_1(x) < 0, h_2(x) < 0\} \quad & R_3 &= \{h_1(x) > 0, h_2(x) < 0\} \end{aligned} \tag{11}$$

And we define the positive/negative portion of $\Sigma_k$ as

$$\Sigma_1^{\pm} = \{x \in \Sigma_1 : \pm h_2(x) > 0\},\ \Sigma_2^{\pm} = \{x \in \Sigma_2 : \pm h_1(x) > 0\}$$

Similarly as in the codimension 1 case, we define the (attractive) sliding region, $\widehat{\Sigma}_k$, of $\Sigma_k$ as the portions where it is finite-time attractive within some neighbourhood, and $\widehat{\Sigma} = \widehat{\Sigma}_1 \cap \widehat{\Sigma}_2$ as the attractive portion of $\Sigma$.

We can see that any trajectory in (10) can be understood as trajectories either in one of the $R_i$, one of the $\widehat{\Sigma}_k$, or in $\widehat{\Sigma}$, concatenated together. The first segment is a usual ODEs, and the second segment can be locally subjected to the analysis of the codimension 1 case. Therefore it is only the third segment, when $x$ is in $\Sigma$ that is difficult to understand. Indeed, if $x \in \widehat{\Sigma}$, then $\Sigma$ is being attractive and so any vector field defined on it should keep trajectories within $\Sigma$ i.e. our Filippov sliding vector field should satisfy:

$$F_s(x) = \sum_{i=1}^{4} \lambda_i(x) f_i(x) \text{ s.t. } \nabla h_k(x)^T F_s(x) = 0\,, k = 1, 2$$

or

$$\begin{pmatrix} w_1^1 & w_2^1 & w_3^1 & w_4^1 \\ w_1^2 & w_2^2 & w_3^2 & w_4^2 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

a system of 3 equations in 4 unknowns. We demonstrate this with a simple example.

**Example 5.13.** Consider the piecewise constant PWS in $\mathbb{R}^3$:

$$f_1 = (1, 1, 1),\ f_2 = (1, -1, v),\ f_3 = (-1, 1, v),\ f_4 = (-1, -1, 1) \tag{12}$$

with $v \leq 1$ and the separating surfaces being the $x$ and $y$ plane, i.e. $h_1(x, y, z) = x$, $h_2(x, y, z) = y$ . We compute $\nabla h_1 = (1, 0, 0)$ and $\nabla h_2 = (0, 1, 0)$ so that

$$w_1 = (1, 1), w_2 = (1, -1), w_3 = (-1, 1), w_4 = (-1, -1).$$

The general Filippov solution will be

$$F(x) = \sum_{i=1}^{4} \lambda_i f_i(x) = \begin{pmatrix} \lambda_1 + \lambda_2 - \lambda_3 - \lambda_4 \\ \lambda_1 - \lambda_2 + \lambda_3 - \lambda_4 \\ \lambda_1 + v\lambda_2 + v\lambda_3 + \lambda_4 \end{pmatrix}$$

Applying the Filippov sliding conditions, $\nabla h_k^T F(x) = 0$, we have the system of equations:

$$\begin{cases} \lambda_1 + \lambda_2 - \lambda_3 - \lambda_4 = 0 \\ \lambda_1 - \lambda_2 + \lambda_3 - \lambda_4 = 0 \\ \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1 \end{cases}$$

the first two lines gives $\lambda_1 = \lambda_4$ and $\lambda_2 = \lambda_3$. Combining with the last, we get $\lambda_2 = \frac{1}{2}(1 - \lambda_4)$. Since we require $\lambda_2 \in [0, 1]$, this implies we must restrict the

range to $\lambda_4 \in [0, \frac{1}{2}]$. Therefore the Filippov sliding solution will be given by

$$F_s(x) = \sum_{i=1}^{4} \lambda_i f_i(x) = \begin{pmatrix} 0 \\ 0 \\ 2\lambda_4(1-v)+v \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \lambda(1-v)+v \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \lambda+(1-\lambda)v \end{pmatrix}$$

where we $\lambda \coloneqq 2\lambda_4 \in [0,1]$. For $\lambda = 0$ we have a $z$-velocity of $v$ (as in $f_{2,3}$), and for $\lambda = 1$ we have a $z$-velocity of 1 (as in $f_{1,4}$). Note if $v = 1$, all vector fields have the same $z$-velocity and so the Filippov sliding mode is unique here. Otherwise it is a one-parameter family of solutions.

Although there are special cases in which this has a singular solution as in the previous example, generally this gives a one-parameter family of solutions and so an ambiguous or set valued vector field on $\widehat{\Sigma}$. For general $p$, we will have the set of Filippov solutions being a $2^p - p - 1$-parameter family of solutions. Thus we are still left with a conundrum for $p \geq 2$, in that the Filippov sliding conditions are not enough to give a unique vector field.

## 5.3 Realisation and Regularisation

We are still interested in finding a single-valued vector field on $\widehat{\Sigma}$, but basic Filippov theory cannot take us any further than a one parameter set of vector fields. As noted in [25], we must impose extra conditions on (10) in order to achieve a single-valued vector field on $\widehat{\Sigma}$. What these extra regularisation conditions should be is entirely determined by the underlying system we are trying to model.

In Euler $\epsilon$-IQL, we used an approximation method to define trajectories of a PWS in order to overcome the discontinuity caused by the fact that $d_\epsilon$ is not well-defined for Q-values with more than one greedy action. There are other possible ways to model how agents would deal with two Q-values with equally high values: perhaps when they are very close they sample actions from a "blended" distribution with peaks at the greedy actions, or only play actions they have recently played until the Q-values are no longer equal and then play the greedy action, or have some kind of inertia against switching actions. Many additional constraints have been considered in the study of PWSs that patch over the discontinuity surface in some way, but still manage to approximate the family of Filippov trajectories within some error $\varepsilon$. Thankfully, there is a theorem from [28] which unites all these approximation methods.

Suppose for attractive $\Sigma$ and a small $\varepsilon > 0$, there is a $\varepsilon$-neighbourhood of $\Sigma$ called $\mathcal{N}_\varepsilon$ where the RHS of (10) is well-defined (i.e. a neighbourhood of $\Sigma$ with $\Sigma_1$, $\Sigma_2$ removed). From within $\mathcal{N}_\varepsilon$, we select one of the $f_i$ to follow[6] over a short constancy time interval according to some method, provided that it causes trajectories to remain within $\mathcal{N}_\varepsilon$. Any such sequence of selections would cause a trajectory $x_\varepsilon$ to zigzag around within $\mathcal{N}_\varepsilon$, much like how Euler $\epsilon$-IQL

[6]Note that the selected $f_i$ does not need to be the one corresponding to which $R_j$ $x_\varepsilon$ finds itself in. This is only the case for the Euler method.

did. This "chattering approximation" $x_\varepsilon$ would behave somewhat like a sliding trajectory; a resultant velocity somewhere in between all of the $f_i$ and sticking to $\Sigma$, albeit within some neighbourhood. Shrinking $\mathcal{N}_\varepsilon$ to $\Sigma$ as $\varepsilon \to 0$ (with increasingly shorter constancy time intervals), we may have a limit of $x_\varepsilon \to x_0$ (uniformly). Since each $x_\varepsilon$ was confined to $\mathcal{N}_\varepsilon$ and $\mathcal{N}_\varepsilon \to \Sigma$, the $x_0$ is one of the Filippov sliding trajectories. Where this occurs, the $\varepsilon$-solution $x_\varepsilon$ is said to "realise" $x_0$.

**Theorem 5.14** (Realisability Theorem)**.** *Every Filippov sliding trajectory is realisable.*

We refer to [28] for a proof. The crucial point in the proof is subdividing the constancy time intervals in a way such that we can weight the strength of each $f_i$ in a similar manner to how the $\lambda_i$s weight the strength of each $f_i$ in Filippov solutions. An similar formulation is found in [29, 30].

**Definition 5.15** (Quasitrajectory)**.** Suppose we have a differential inclusion $\dot{x} \in G(t,x)$ with a function $G(t,x)$ that is continuous in $t$ and $x$ and $G$ such that $\left\|F(t,x)\right\| < m \ \forall t,x$. a quasitrajectory is an absolutely continuous function $z(t)$, $t \in [0,\tau]$ for which there exists a sequence of absolutely continuous functions $x_k(t)$ with the properties

$$\left\|\dot{x}_k(t)\right\| \leq m; \quad x_k(t) \to z(t) \quad t \in [0,\tau]$$

and $\rho(\dot{x}_k(t), G(t, \dot{x}_k(t))) \to 0$ almost everywhere on $[0,\tau]$. Here, $\rho$ is the Hausdorff distance: $\rho(x,A) = \inf\limits_{a\in A} d(x,a)$

Then a similar theorem to Theorem 5.14 is

**Theorem 5.16.** *Let $Q(G)$ denote the set of quasitrajectories of a differential inclusion with right hand side $G(t,x)$, such that $G$ is non-empty, bounded and closed, and is continuous in $t$ and $x$. Let $S(\overline{\text{co}}\, G)$ denote the set of all solutions to the differential inclusion $\dot{x}(t) \in \overline{\text{co}}\, G(t,x)$. Then*

$$Q(G) = S(\overline{\text{co}}\, G)$$

*i.e. trajectories of continuous, closed, bounded and convex differential inclusions can be realised as uniform limits of approximate solutions.*

Theorem 5.16 essentially says that for any approximation method which gives trajectories with derivatives $\varepsilon$ close to the set-valued RHS almost everywhere, then as $\varepsilon \to 0$ our approximation method gives a (quasi)trajectory which solves the system. Theorem 5.14 strengthens this to say that when $\Sigma$ is attractive the quasitrajectory will also be a sliding trajectory. Indeed, one proof of Theorem 5.5 approximates the RHS using an Euler polygon, so the limit of any trajectory created by the Euler method is a Filippov solution. Thus we can see why Proposition 5.2 holds: since Euler polygons approximate Filippov sliding trajectories, and in codimension $p = 1$ the set of Filippov sliding trajectories is

singular, the limit of the Euler method is unique and realises the sliding trajectory. One thing to note though is that when the set of Filippov sliding solutions is not singular, neither need be the limit of the Euler polygons. We may get different sliding solutions depending on the subsequence we descend $h \to 0$ by, as in Section 5.6.

Although the above theorems tells us that any reasonable approximation method is a Filippov sliding trajectory and that any Filippov sliding trajectory can be reached by using some approximation, it does not tell us which class of solutions we achieve if we restrict ourselves to just one approximation method. For Euler $\epsilon$-IQL therefore, we do not know what Filippov sliding solution, or set of Filippov sliding solutions, we will obtain in the Opportunistic Defection phase as $h \to 0$. I am not aware of any methods in the literature that would be able to answer this, as the Euler method applied in general codimension 2 PWSs seems very complex. There are error estimates on the Hausdorff distance between the set of solutions of a differential inclusion and the set of Euler approximate solutions given in [31, 32], which could perhaps set bounds on the speed at which players slide in the Opportunistic Defection phase.

For the remainder of this report, we shall explore some of these reasonable or "natural" approximation methods. As noted in [28], any Filippov trajectory can be realised as a limit of $x_\varepsilon$ as $\varepsilon \to 0$ where $x_\varepsilon$ is specially constructed to precisely fulfill this limit. A more interesting path is to to consider reasonable regularisations of (10) as additional mechanisms to consider, and find which solutions these realise. Many such regularisation methods already exist in the literature, some of which we list below.

(a) Hysteresis [33]: Define a "chatterbox" $C_\varepsilon$ around $\Sigma$ as

$$C_\varepsilon \coloneqq \{x \in \mathbb{R}^n : \left|h_1(x)\right|, \left|h_2(x)\right| < \varepsilon\}$$

When a trajectory approaches $\Sigma$ from $R_i$, we continue to use $f_i$ until hitting the boundary of $C_\varepsilon$.

(b) Sigmoid Blending [28]: Interpolate/Blend the $f_i$s within the chatterbox via sigmoid curves.

(c) Stochastic Switching: When a trajectory approaches $\Sigma$ from $R_i$, use $f_i$ but switch to another $f_j$ according to some Poisson process on whether we take $h_k$ to be positive or negative, and switches with certainty on the boundary of $C_\varepsilon$.

(d) Stochastic DE: Add Brownian noise to (10) i.e. replace $\dot{x} = f_i(x)$ for $x \in R_i$ with

$$\dot{x} = f_i(x)dt + \varepsilon dw_t$$

Note this will land on $\Sigma$ with probability zero.

(e) Forward Euler: When approaching $\Sigma$ from $R_i$, continue to use $f_i$ for a constant time step $\varepsilon$.

(f) Some combination of all of the above.

All of these (apart from (c) and (d)) effectively replace the PWS (10) with a new (possibly discrete) dynamical system, regularised by the error term $\varepsilon$. In codimension $p = 1$, all will realise the unique sliding trajectory. In codimension $p = 2$ and higher however, each will have their own unique dynamics, and so their deployment should reflect some mechanical interpretation of the underlying system.

One of the most important differences between different regularisation techniques is that of exit conditions. If a sliding trajectory is sliding within $\Sigma$, but $\Sigma$ ceases to be attractive, should the trajectory leave $\Sigma$? If the inequality in Definition 5.9 becomes an equality for a neighbourhood of $\Sigma$, but does not become a transverse crossing, should the trajectory stay within $\Sigma$, or exit anyway? This turns out to be a very complicated questions and is determined by the amount of smoothness we wish to enforce on our trajectories. If we require trajectories to be differentiable, the exit points are called first order exit points. As noted in [25], different regularisation methods give very different behaviour at exit points. We shall not consider exit points due to its complexity, but it would be a very fruitful area of study in helping to understand when agents leave the Opportunistic Defection phase in Euler $\epsilon$-IQL on the Prisoner's Dilemma and figuring out if the motion there is truly chaotic or not.

In what follows, we shall expand on some approximation/regularisation methods for codimension $p = 2$. As their approximation parameter $\varepsilon \to 0$, some methods only realise 1 Filippov solution (as for bilinear), some many Filippov solutions (as for Euler) and some are conjectured to give every possible Filippov solution (as for time delay).

## 5.4 Bilinear Method

This was proposed in [28] as an extension of the "fuzzification" intuition we have of the Filippov vector field in codimension $p = 1$. This restriction can be seen as a limit of the sigmoid blending technique as $\varepsilon \to 0$. Within the chatterbox, we interpolate between $R_1$ and $R_3$, and $R_2$ and $R_4$ by a smooth sigmoid function $s_1(x)$ such that for $x \in \partial C_\varepsilon \cap \{h_1(x) > 0\}$ $s_1(x) = 1$ and for $x \in \partial C_\varepsilon \cap \{h_1(x) < 0\}$ $s_1(x) = 0$. We interpolate between $R_1$ and $R_2$, and $R_3$ and $R_4$ with another sigmoid function $s_2(x)$ with values the same as for $s_1$ but replacing $h_1$ with $h_2$. Then we write the new spatially-regularised and singular valued differential equation as

$$\begin{aligned}\dot{x} =& \frac{1-s_1(x)}{2}\frac{1-s_2(x)}{2}f_1(x) + \frac{1-s_1(x)}{2}\frac{1+s_2(x)}{2}f_2(x) \\ &+ \frac{1+s_1(x)}{2}\frac{1-s_2(x)}{2}f_3(x) + \frac{1+s_1(x)}{2}\frac{1+s_2(x)}{2}f_4(x)\end{aligned} \tag{13}$$

Taking $\varepsilon \to 0$, we interpret $s_1(x)$ and $s_2(x)$ becoming multivalued at $h_1(x)$ and $h_2(x)$ respectively i.e. for $x \in \Sigma_k$, $s_k(x) = [0, 1]$. Thus the limiting vector field

is the set-valued one [27]

$$F_B(x) = (1-\alpha)(1-\beta)f_1 + (1-\alpha)\beta f_2 + \alpha(1-\beta)f_3 + \alpha\beta f_4,\ \alpha,\beta \in [0,1] \quad (14)$$

where we write $s_1(x) = 2\alpha(x) - 1$ and $s_2(x) = 2\beta(x) - 1$. We shall omit the dependence of $\alpha, \beta$ on $x$ where it is obvious. This method effectively sidesteps the problem of non-uniqueness of the $\lambda_i$s, by reducing the class of Filippov vector fields admissible on $\widehat{\Sigma}$ via further constraints on our $\lambda_i$s, so that we get some unique Filippov sliding vector field.

On $\widehat{\Sigma}$, we will require $\nabla h_k^T F_B(x) = 0$ for it to be a Filippov sliding vector field. By rewriting (14), we can see any $\alpha, \beta$ must satisfy the non-linear system

$$(1-\alpha)(1-\beta)\begin{pmatrix} w_1^1 \\ w_1^2 \end{pmatrix} + (1-\alpha)\beta\begin{pmatrix} w_2^1 \\ w_2^2 \end{pmatrix} + \alpha(1-\beta)\begin{pmatrix} w_3^1 \\ w_3^2 \end{pmatrix} + \alpha\beta\begin{pmatrix} w_4^1 \\ w_4^2 \end{pmatrix} = 0 \quad (15)$$

The solvability of this system was analysed extensively in [27]. Dieci, Elia and Lopez proved that under Assumptions 5.17 on the $w_i^k(x)$ in a neighbourhood of $\Sigma$, there exists unique "fuzzy truth values" $(\bar{\alpha}, \bar{\beta}) \in (0,1)^2$ that solve 15 that give us a Filippov sliding vector field, and vary smoothly with $x$.

**Assumptions 5.17.** (a) $(w_i^1(x), w_i^2(x))$ does not have identical component wise sign to $(h_1(x), h_2(x))$ for $x \in R_i, i = 1,2,3,4$

(b) At least one pair of the following relations ($(1^+)$ and $(1_a^+)$ for example) is satisfied on $\Sigma$ and in a neighbourhood of $\Sigma$:

$$\begin{aligned}
&(1^+) \quad w_2^1 > 0, w_4^1 < 0, && (1_a^+) \quad \frac{w_2^2}{w_2^1} - \frac{w_4^2}{w_4^1} < 0,\\
&(1^-) \quad w_1^1 > 0, w_3^1 < 0, && (1_a^-) \quad \frac{w_3^2}{w_3^1} - \frac{w_1^2}{w_1^1} < 0,\\
&(2^+) \quad w_3^2 > 0, w_4^2 < 0, && (2_a^+) \quad \frac{w_3^1}{w_3^2} - \frac{w_4^1}{w_4^2} < 0,\\
&(2^-) \quad w_1^2 > 0, w_2^2 < 0, && (2_a^-) \quad \frac{w_2^1}{w_2^2} - \frac{w_1^1}{w_1^2} < 0
\end{aligned}$$

(c) If any of $(1^\pm)$ or $(2^\pm)$ is satisfied, then $(1_a^\pm)$ or $(2_a^\pm)$ is satisfied also.

(a) guarantees that the vector fields $f_i$ must all not point totally away from $\Sigma$ and towards at least one of $\Sigma_{1,2}$. (b) guarantees that attractive sliding towards $\Sigma$ occurs on at least one of $\Sigma_{1,2}^\pm$. (c) guarantees that if attractive sliding occurs along $\Sigma_{1,2}^\pm$, then it must be towards $\Sigma$. The PWS shown in Figure 28 would satisfy the above assumptions, but the PWS shown in Figure 23 would not as $f_1$ in this example has trajectories which are not eventually attracted to $\Sigma$.
These assumptions do not actually rule out the case of repulsive sliding, but do require that any sliding on one of $\widehat{\Sigma}_{1,2}$ is towards $\Sigma$, so that every trajectory in a neighbourhood of $\widehat{\Sigma}$ is still attracted towards $\widehat{\Sigma}$ in finite time.

The proof involves various rewritings of (15) in order to give us 2 quadratic polynomials $P(\beta)$ and $Q(\alpha)$, with roots $(\bar{\alpha}, \bar{\beta}) \in (0,1)^2$ (as well as functions $g(\beta)$ and $h(\alpha)$ such that $g(\bar{\beta}) = \bar{\alpha}$, $h(\bar{\alpha}) = \bar{\beta}$. Due to the Assumptions 5.17 and several applications of the Intermediate Value Theorem, only one of the roots of $P$ will remain in $(0,1)$ under $g$ (and the same for the roots of $Q$ and $h$). The expressions for these are listed below:

$$\begin{aligned} P(\beta) =&[(1-\beta)w_1^2 + \beta w_2^2][(1-\beta)w_3^1 + \beta w_4^1] \\ &-[(1-\beta)w_3^2 + \beta w_4^2][(1-\beta)w_1^1 + \beta w_2^1] \end{aligned} \tag{16}$$

$$g(\beta) = \frac{(1-\beta)w_1^2 + \beta w_2^2}{[(1-\beta)w_1^2 + \beta w_2^2] - [(1-\beta)w_3^2 + \beta w_4^2]} \tag{17}$$

$$\begin{aligned} Q(\alpha) =&[(1-\alpha)w_1^1 + \alpha w_3^1][(1-\alpha)w_2^2 + \alpha w_4^2] \\ &-[(1-\alpha)w_2^1 + \alpha w_4^1][(1-\alpha)w_1^2 + \alpha w_3^2] \end{aligned} \tag{18}$$

$$h(\alpha) = \frac{(1-\alpha)w_1^1 + \alpha w_3^1}{[(1-\alpha)w_1^1 + \alpha w_3^1] - [(1-\alpha)w_2^1 + \alpha w_4^1]} \tag{19}$$

To prove that $P$ and $Q$ have roots in $(0,1)$, one then has to consider every possible case for Assumptions 5.17 (b,c) i.e. every combination of whether or not there is attractive sliding along each of $\Sigma_1^+$, $\Sigma_1^-$, $\Sigma_2^+$ or $\Sigma_2^-$ of which there are 13 in total (up to equivalence). For now we shall see a simple example.

**Example 5.18.** Consider the piecewise constant PWS in $\mathbb{R}^3$ from Example 5.13:

$$f_1 = (1,1,1),\ f_2 = (1,-1,v),\ f_3 = (-1,1,v),\ f_4 = (-1,-1,1) \tag{20}$$

with $v \leq 1$ and the separating surfaces being the $x$ and $y$ plane, i.e. $h_1(x,y,z) = x$, $h_2(x,y,z) = y$. We compute $\nabla h_1 = (1,0,0)$ and $\nabla h_2 = (0,1,0)$ so that

$$w_1 = (1,1), w_2 = (1,-1), w_3 = (-1,1), w_4 = (-1,-1).$$

This PWS trivially satisfies Assumptions 5.17 due to it being a nodally attractive case. Although generally we would look for roots of $P$ and $Q$, in this case we can (15) as

$$(1-\alpha)(1-\beta)\begin{pmatrix}1\\1\end{pmatrix} + (1-\alpha)\beta\begin{pmatrix}1\\-1\end{pmatrix} + \alpha(1-\beta)\begin{pmatrix}-1\\1\end{pmatrix} + \alpha\beta\begin{pmatrix}-1\\-1\end{pmatrix} = 0$$

which simplifies to

$$\begin{pmatrix}1-2\alpha\\1-2\beta\end{pmatrix} = 0$$

i.e. $\alpha = \beta = \frac{1}{2}$. Thus our Filippov sliding solution is given by $F_B(x) = \begin{pmatrix}0\\0\\\frac{v+1}{2}\end{pmatrix}$ so that sliding trajectories with initial condition $x_0 \in \Sigma$ will be given

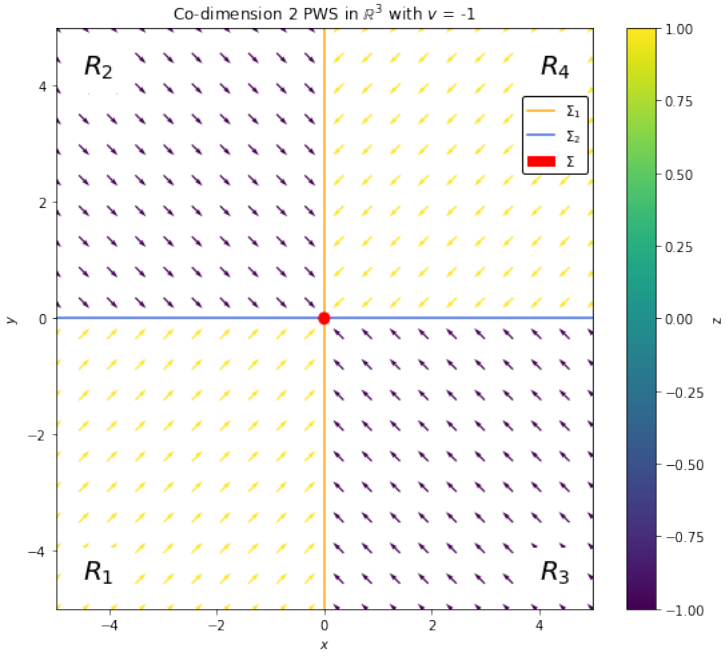


Figure 28: Plot of the PWS (12) with $v = -1$.

by $x(t) = x_0 + (0, 0, t(\frac{v+1}{2}))$. Note that the $F_B$ ended up being an average of the surrounding fields. See Figure 29 for a graph of the Discrete Bilinear method described in 6.1.2, which seems to agree with the above result.

Since $F_B$ considers only the vector fields around $\Sigma$, without any information from previous behaviour of trajectories, it is usually refered to as a space regularisation. Other spatial regularisations that seek to impose additional algebraic constraints in order to select a singular Filippov sliding vector are possible of course, such as the moments method, introduced in [34]. Several things should be considered before adding extra algebraic constraints such as the bilinear model to any dynamical system. The bilinear model may not have use in the PWS model of $\epsilon$-IQL; we would have to say that our agents would have some fuzzy idea of their Q-values. Although this would give us a fully defined vector field on the indifference plane, it is quite a strong assumption to place on the capabilities of our agents. Due to the fact it is a space regularisation, it involves a direct manipulation of the PWS rather than merely "looking at what is there" as in Euler method. Thus a more appropriate regularisation may be one that does not alter how the agents act, but rather how they react: regularising time as opposed to space.

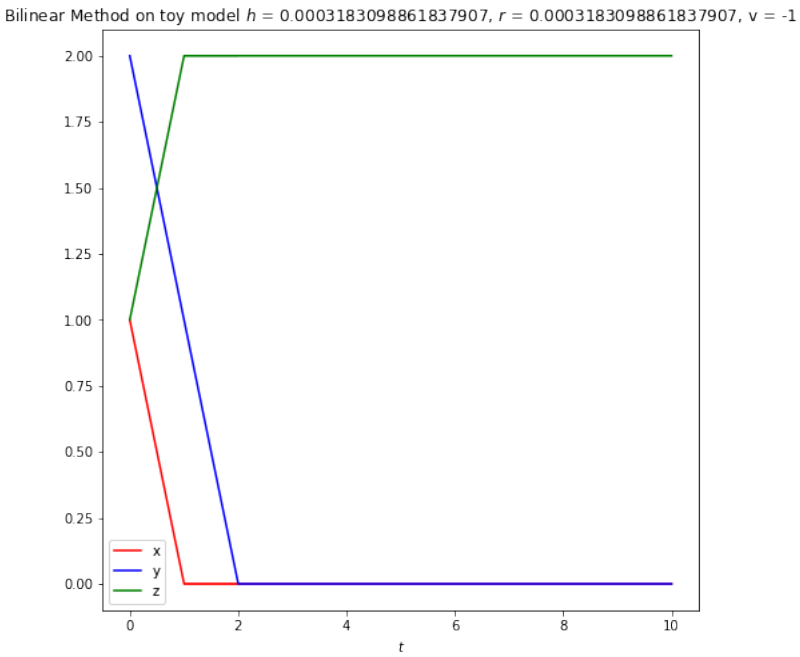


Figure 29: Discrete Bilinear method as applied to Example 5.18. The gradient of the $z-$component is 0, agreeing with the result.

## 5.5 Time Delay Method

Bilinear method was a local regularisation technique: we defined a chatterbox neighbourhood of $\Sigma$ in which the dynamics was altered. The time-averaging or time delay method, introduced in [35], is instead a global regularisation as well as a temporal one. In this method, we aim to avoid the discontinuity at the $\Sigma_k$s by completely redefining how motion is defined in the PWS, similar to Euler method. This method exploits the fact that $\Sigma_k$ is a null set by adding a kind of inertia to particles by a differential integral equation, in which we follow the average velocity over the last $\delta$ time units.

### 5.5.1 Codimension $p = 1$ case

The codimension $p = 1$ case is very well understood and realises Filippov sliding trajectories. We replace the codimension $p = 1$ PWS with an initial value on the discontinuity surface:

$$\dot{x} = f(x), f(x) = \begin{cases} f_+(x), & \text{for } h(x) > 0 \\ f_-(x), & \text{for } h(x) < 0 \end{cases} \tag{21}$$

$$x(0) = x_0 \in \Sigma_1 = \Sigma$$

with the approximated problem

$$\begin{cases} \dot{x}(t) = \frac{1}{\delta} \int_{t-\delta}^{t} \left[ H(h(x(s)))f_+(x(s)) + H(-h(x(s)))f_-(x(s)) \right] ds, \ t > 0 \\ x(t) = \psi(t), \ t \in [-\delta, 0] \end{cases} \tag{22}$$

where $H$ is the Heaviside function[7] and the initial trajectory $\psi$ is any continuous function that satisfies

$$\begin{cases} \psi(0) = x_0 \in \Sigma \\ h(\psi(t)) \neq 0, t \in [-\delta, 0) \\ \left| \psi(t) - \psi(0) \right| \leq M|t|, t \in [-\rho, 0] \text{ for some } M, \rho > 0. \end{cases} \tag{23}$$

Equation (22) is replacing the RHS of (21) with the average of its velocity in the time interval $[t - \delta, t]$. This gives a new time delay differential equation (as opposed to the set-valued differential equation of the Filippov solutions) that is defined even when $x(s) \in \Sigma$ (given (23)): as we will see, the solution will oscillate/chatter around $\Sigma$ in a similar manner to the Euler method. It will only be on $\Sigma$ for a null set of time, which is why it is essentially discarded when taking the integral. See Figures 30 and 31 for an example of what motion looks like under the Time Delay method.

[7] $H : \mathbb{R} \to \{0, 1\}$ such that $H(x) = 1$ for $x \geq 0$ and $H(x) = 0$ for $x < 0$. Whatever convention of the Heaviside function we use, the difference will be lost in the integral.

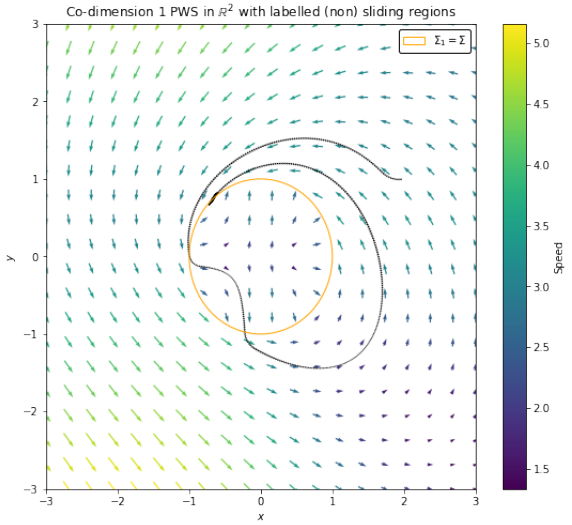


Figure 30: Plot of a trajectory following the time delay method in a PWS of codimension 1. Notice how after transversely crossing $\Sigma$ a few times, it eventually reaches a sliding region and begins oscillating around it, before appearing to reach a fixed point.

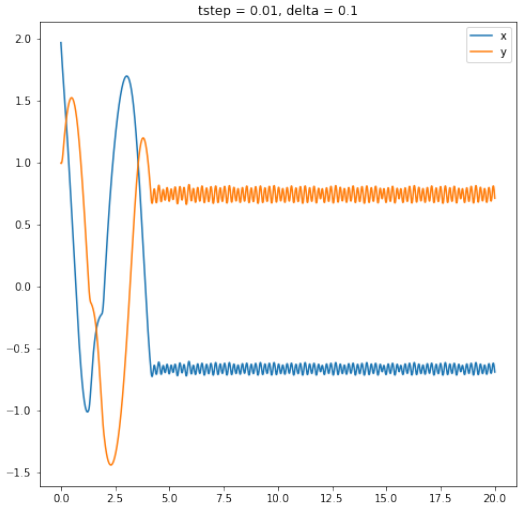


Figure 31: Plot of $x$ and $y$ values against time of the trajectory following the time delay method in the PWS shown in Figure 30. Note how after reaching a sliding region it oscillates almost sinusoidally around $\Sigma$.

**Theorem 5.19** (Fusco & Guglielmi)**.** *Let $f_+, f_-, \psi$ satisfy the conditions above, with $f_+$ and $f_-$ both pointing towards $\Sigma$ (attractive). Then there exists $\delta_0 > 0, T > 0$ such that for each $\delta \in (0, \delta_0)$, (22) has a $C^1$-solution $x^\delta : [0,T] \to \mathbb{R}^n$. Moreover:*

1. $\left|h(x^\delta(t))\right| \leq C\delta \ \forall t \in [0,T]$

*There is a $C^1$ function $x^0$ such that:*

2. $h(x^0(t)) = 0 \, \forall t \in [0,T]$
3. $\lim_{\delta \to 0} \left\| x^\delta - x^0 \right\|_{C^0([0,T])} = 0$
4. *$x^0$ is a Filippov sliding vector field solution:*

$$\dot{x} = F_s(x) \in T_x\Sigma, \ x(0) = x_0 \in \Sigma$$

*where $F_s(x)$ is given as in Lemma 5.1.*

Moreover, any $\psi$ satisfying (23) will give the same solution in the singular limit $\delta \to 0$.

In the proof of Theorem 5.19, it is assumed $h(x) = x_n$ so that $\Sigma$ is a plane, but any $\Sigma$ can be flattened to this case by the Implicit Function Theorem. It is also assumed that the vector fields are constant in a neighbourhood of $\Sigma$: $f_+(x) = p \leq 0$, $f_-(x) = q \geq 0$. Now the problem becomes:

$$\begin{cases} \dot{x}(t) = \frac{1}{\delta} \int_{t-\delta}^{t} \left[ H(x_n(s))p + H(-x_n(s))q \right] ds, & t > 0 \\ x(t) = \psi(t), & t \in [-\delta, 0] \end{cases} \tag{24}$$

We reparameterise:

$$x(t) = \psi(0) + \delta\xi(\tau), \ t = \delta\tau$$

to give:

$$\begin{cases} \dot{\xi}(\tau) = \int_{\tau-1}^{\tau} \left[ H(\xi_n(s))p + H(-\xi_n(s))q \right] ds, & \tau > 0 \\ \xi(\tau) = \varphi(\tau) = \frac{1}{\delta}(\psi(\delta\tau) - \psi(0)) & \tau \in [-1, 0] \end{cases} \tag{25}$$

The solution $\xi$ is $C^{1,1}$ and the $n$th component $\xi_n$ oscillates around $\{y_n = 0\}$ with infinitely many (isolated) zeros $0 < \tau_1 < \tau_2 < \ldots$. In fact, $\xi$ is periodic and has an analytic solution:

**Lemma 5.3.** *For $\varphi_n(\tau) > 0$ for $\tau \in (-1, 0)$, there exists a $C^{1,1}$ function $\eta : \mathbb{R} \to \mathbb{R}$ that is 2-periodic and*

$$\xi_n(\tau) = \eta\left( \tau - \tau_1 - \left\lfloor \frac{\tau - \tau_1}{2} \right\rfloor \right), \ \text{for } \tau \geq \tau_1$$

*where $\tau_1 = 2\bar{\mu}$ and $\bar{\mu} = \frac{p_n}{p_n - q_n}$*[8]*.*

[8] If $\bar{\mu} > \frac{1}{2}$, we can swap $f_+$ and $f_-$ and $h$ with $-h$ so that the equation for $\eta$ is well-defined.

Moreover, $\eta : \mathbb{R} \to \mathbb{R}$ can be expressed as (continued 2-periodically)

$$\eta(\tau) = \begin{cases} -p_n\tau, & \tau \in [0, 1-2\bar{\mu}] \\ -p_n\tau + \frac{1}{2}(p_n - q_n)(\tau - 1 + 2\bar{\mu})^2, & \tau \in [1-2\bar{\mu}, 1] \\ (2\bar{\mu}-1)p_n + p_n(\tau - 1), & \tau \in [1, 2(1-\bar{\mu})] \\ \frac{q_n - p_n}{2}(\tau - 2)(\tau - 2(1-\bar{\mu})), & \tau \in [2(1-\bar{\mu}), 2] \end{cases}$$

Thus we have a kind of "fundamental oscillation" around the separating surface. By reparameterising, we get an analytic solution to (24) that is $2\delta$-periodic. For details of how we generalise this to non-constant vector fields we refer to [35]. Thus the codimension $p = 1$ case of the Time Delay method is well understood, and gives an approximation that converges to sliding solutions and this approximation is given by a periodic function that can be derived and written in a closed form. It was due to the similar oscillatory behaviour of the Time Delay method that reminded me of the Euler method's behaviour that made me think there was some connection between the two, perhaps a value of $\delta$ related to $h$ that would cause Time Delay to have identical behaviour, as in Example 5.20.

### 5.5.2 Codimension $p = 2$ case

For a PWS with separating surface $\Sigma$ of codimension $p = 2$, the behaviour of the time delay method is not as well understood. Since the Filippov vector field is undetermined and usually non-unique, correspondingly the solution $x^\delta$ does not converge to a single solution. If we can imagine a trajectory oscillating around, say $\Sigma_1$, and moving towards $\Sigma$, when the trajectory crosses $\Sigma_2$, it will also begin oscillating around $\Sigma_2$. The phase difference of these two oscillations will determine the time spent in each of the $R_i$ and determine the overall motion. However, the phase difference depends greatly on the $\delta$ parameter and so how the $\delta$ goes to zero changes the resultant motion.

Although a rigorous and general analysis of the behaviour of $x^\delta$ once it reaches an $\delta$-neighbourhood of $\Sigma$ has not yet been completed, from [35], we have the conjecture (here $d(x, A) = \inf_{y \in A} d(x, y)$, the Hausdorff distance):

**Conjecture 5.4.** *The distance $d(x^\delta(t), \Sigma)$ satisfies*

$$d(x^\delta(t), \Sigma) \leq C\delta \text{ for } t \in [\bar{t}, \bar{t} + \Delta),$$

*where $C, \Delta > 0$ independent of $\delta$ and $\bar{t}$ is the least time when $x^\delta(t)$ enters a $\delta$-neighbourhood of $\Sigma$. Furthermore,*

*(i) If the Filippov sliding vector field on $\Sigma$ is unique, then there exists*

$$x^0(t) = \lim_{\delta \to 0^+} x^\delta(t) \text{ for } t \in [\bar{t}, \bar{t} + \Delta),$$

*where $x^0$ is the Filippov sliding solution lying on $\Sigma$ with initial condition $\bar{x}$.*

*(ii) If the Filippov sliding vector field on $\Sigma$ is not unique, then $x^\delta(t)$ converges only along subsequences $\delta_n \to 0^+$. Moreover given any Filippov sliding vector field $F(\bar{x}) \in T_{\bar{x}}\Sigma$, the sequence $\delta_n$ can be chosen in such a way that*

$$\frac{d}{dt} \lim_{\delta_n \to 0^+} x^{\delta_n}(\bar{t}) = F(\bar{x})$$

**Example 5.20.** Consider again the piecewise constant PWS in $\mathbb{R}^3$ from Example 5.13:

$$f_1 = (1, 1, 1),\ f_2 = (1, -1, v),\ f_3 = (-1, 1, v),\ f_4 = (-1, -1, 1) \tag{26}$$

with $v \leq 1$ and the separating surfaces being the $x$ and $y$ plane, i.e. $h_1(x, y, z) = x$, $h_2(x, y, z) = y$, with initial condition $x_0 = (a, b, c) \in R_4$ such that $b > a > 0$, so that we are guaranteed to hit $\Sigma_1$ first. We know that the set of Filippov sliding solutions is given by

$$F_s(x) = \begin{pmatrix} 0 \\ 0 \\ \lambda(1 - v) + v \end{pmatrix}, \quad \lambda \in [0, 1]$$

We will aim to show that there exists a subsequence $\delta_n \to 0$ such that the approximate solution $\dot{x}^{\delta_n}$ converges to anyone of these Filippov sliding solutions. The regularised problem is:

$$\begin{aligned}\dot{x}(t) = \frac{1}{\delta} \int_{t-\delta}^{t} & H(h_1(x(s)))H(h_2(x(s)))f_4 + H(h_1(x(s)))H(-h_2(x(s)))f_3 \\ & + H(-h_1(x(s)))H(h_2(x(s)))f_2 + H(-h_1(x(s)))H(-h_2(x(s)))f_1 ds\end{aligned}$$

Using the fact that $H(a) + H(-a) = 1\, \forall a \in \mathbb{R}$, and writing $H_i = H(h_i(x(s)))$ this becomes

$$\dot{x}(t) = \frac{1}{\delta} \int_{t-\delta}^{t} H_1 H_2(f_1 - f_2 - f_3 + f_4) + H_1(f_3 - f_1) + H_2(f_2 - f_1) + f_1 ds$$

which component wise reduces to:

$$\dot{x}_1(t) = \frac{1}{\delta} \int_{t-\delta}^{t} 1 - 2H_1 ds \tag{27}$$

$$\dot{x}_2(t) = \frac{1}{\delta} \int_{t-\delta}^{t} 1 - 2H_2 ds \tag{28}$$

$$\dot{x}_3(t) = \frac{1}{\delta} \int_{t-\delta}^{t} (1 - v)[2H_1 H_2 - H_1 - H_2] + 1 ds \tag{29}$$

Note that the equations for $\dot{x}_1$ and $\dot{x}_2$ are decoupled from each other, so in essence this reduces to the codimension 1 case for those components. This rarely true for general piecewise constant vector and relies on the high amount

of symmetry in this particular example. The $x_3$ component is more complicated, depending on both of these for the $v \neq 1$ case.

The initial datum $\psi(t)$ which defines the solution in $[-\delta, 0]$ is defined as $\psi(t) = (a, b, c)$, so that it is independent of $\delta$ and $\psi([-\delta, 0)) \subset R_4$. It could equivalently be defined as $\psi(t) = (a-t, b-t, c+t)$, as this example is a piecewise constant vector field. We will see that there are 3 regimes: approaching $\Sigma_1$ from $R_4$, approaching $\Sigma$ from $\Sigma_1$, and oscillating around $\Sigma$.

If $x(s) \in R_4$ for $s \in [0, t_1]$ for some $t_1$, then $H_1 = H(h_1(x(s))) = 1$ (and similarly for $H_2$) giving

$$\begin{aligned}
\dot{x}_1(t) &= \frac{1}{\delta}\int_{t-\delta}^{t} -1 ds = \frac{1}{\delta}[s]_{t-\delta}^{t} = \frac{1}{\delta}[-t + t - \delta] = -1 \\
\dot{x}_2(t) &= \frac{1}{\delta}\int_{t-\delta}^{t} -1 ds = -1 \\
\dot{x}_3(t) &= \frac{1}{\delta}\int_{t-\delta}^{t} +1 ds = +1
\end{aligned}$$

or $\dot{x}(t) = f_4$ for $t \in [0, t_1]$, so the above integral differential equation has the classical solution $x(t) = (a - t, b - t, c + t)$ for $t \in [0, t_1]$. Since $h_1(x(t)) = 0$ at $t = a$, we can see $t_1 = a$.

$$\text{For } t \in [0, a],\ x(t) = (a - t, b - t, c + t)$$

Let $t_2$ be the least time for which $h_2(x(t)) = 0$. For $t \in [t_1, t_2]$, $h_2(x(t)) > 0$ and so the differential equation for $x_2(t)$ will remain the classical one i.e. $\dot{x}_2(t) = -1$ in this regime, which gives $t_2 = b$. Since the equations for $\dot{x}_1$ and $\dot{x}_2$ are decoupled, $x_2$ will be independent of the behaviour of $x_1$. Note that both $t_1$ and $t_2$ are independent of $\delta$ as well, so the first part of Conjecture 5.4 holds with $\bar{t} = t_2$.

While $x_2$ is descending towards $\Sigma_2$, $x_1(t)$ will begin oscillating around $\Sigma_1$ after $t = t_1$, essentially reducing to the codimension 1 case; we can compute $\bar{\mu} = -\frac{p_1}{q_1 - p_1} = -\frac{-1}{1--1} = \frac{1}{2}$. Note $1 - 2\bar{\mu} = 0$, $[0, 1 - 2\bar{\mu}] = \{0\}$ and $[1, 2(1 - \bar{\mu})] = \{1\}$ meaning the we can ignore the first and third time domains for $\eta(\tau)$. Therefore the fundamental oscillation is

$$\eta(\tau) = \begin{cases} -(-1)\tau + \frac{1}{2}((-1) - 1)(\tau - 1 + 2 \cdot \frac{1}{2})^2 = \tau(1 - \tau), & \tau \in [0, 1] \\ \frac{1-(-1)}{2}(\tau - 2)(\tau - 2(1 - \frac{1}{2})) = (\tau - 1)(\tau - 2), & \tau \in [1, 2] \end{cases} \tag{30}$$

or in terms of $t$, using $x_1(t) = \delta\eta(\tau), t - t_1 = \delta\tau$ and continued $2\delta$-periodically:

$$x_1(t) = \begin{cases} \delta \cdot \frac{1}{\delta}(t - t_1)(1 - \frac{1}{\delta}(t - t_1)) = -\frac{1}{\delta}(t - t_1)(t - t_1 - \delta), & t \in [t_1, t_1 + \delta] \\ \frac{1}{\delta}(\frac{1}{\delta}(t - t_1) - 1)(\frac{1}{\delta}(t - t_1) - 2) = \frac{1}{\delta}(t - t_1 - \delta)(t - t_1 - 2\delta), & t \in [t_1 + \delta, t_1 + 2\delta] \end{cases} \tag{31}$$

extended $2\delta$-periodically. So in this regime, after the particle has hit the $y$-axis but before it hits the $x$-axis, the $x_1$ component will be oscillating around the $y$-axis according to the function above, while $x_2(t)$ will be decreasing with speed $-1$ as in the classical solution. Meanwhile, because $x$ is oscillating between $R_4$ and $R_2$, the $z$-component of the vector field will be switching between 1 and $v$. To see this, note that for $t \in [t_1, t_2]$, $H_1 = 1, H_2 = 0$ meaning that

$$\dot{x}_3(t) = \frac{1}{\delta}\int_{t-\delta}^{t}(1-v)[2H_1H_1-H_1-H_2]+1ds = \frac{1}{\delta}\int_{t-\delta}^{t}(1-v)H_1+vds = \frac{1}{2}(1-v)(1-\dot{x}_1(t))$$

Therefore $x_3$ has oscillatory behaviour in a similar fashion to $x_1$, with identical frequency.

When $x$ hits the $x$-axis at time $t_2 = b$, $x_2$ will begin oscillating around $\Sigma_1$ in a similar fashion to how $x_1$ was oscillating around $\Sigma_2$:

$$x_2(t) = \begin{cases} -\frac{1}{\delta}(t-t_2)(t-t_2-\delta), & t \in [t_2, t_2+\delta] \\ \frac{1}{\delta}(t-t_2-\delta)(t-t_2-2\delta), & t \in [t_2+\delta, t_2+2\delta] \end{cases} \tag{32}$$

extended $2\delta$-periodically. It is important to point out that this oscillation has the same period and amplitude as the $x_1$ oscillation. $x_2$'s oscillating will not affect the motion of $x_1$ as they are decoupled, but it does make the motion of the $x_3$ component more complicated as now the integrand $(1-v)[2H_1H_2-H_1-H_2]+1$ will be switching rapidly between $v$ and $+1$. However, for the $v = 1$ case we have:

$$\dot{x}_3(t) = \frac{1}{\delta}\int_{t-\delta}^{t}(1-1)[2H_1H_2 - H_1 - H_2] + 1ds = \frac{1}{\delta}[t-(t-\delta)] = 1$$

meaning the solution agrees with the Filippov vector field for this value of $v$, at least for the $z$-component. For general $v$, we can exploit the symmetry in this integrand to find in general

$$\dot{x}_3(t) = \begin{cases} 1 + 2\phi(v-1), & \text{for } \phi \in [0, \frac{1}{2}] \\ -1 + 2(\phi + v - v\phi), & \text{for } \phi \in [\frac{1}{2}, 1] \end{cases} \tag{33}$$

with $\phi \in [0, 1)$ defined such that $t_2 = t_1 + 2m\delta + 2\phi\delta$, $m \in \mathbb{Z}_{\geq 0}$. Assume $\phi \in [0, \frac{1}{2}]$. See Figure 32.

Say $t \in [t_1 + \delta + 2\delta\phi + 2\delta m, t_1 + 2\delta + 2\delta m]$. Then

$$t - \delta \in [t_1 + 2\delta\phi + 2\delta m, t_1 + \delta + 2\delta m]$$

Therefore

$$\begin{aligned}[t-\delta, t] = &[t-\delta, t_1 + \delta + 2\delta m] \\ &\cup [t_1 + \delta + 2\delta m, t_1 + \delta + 2\delta\phi + 2\delta m] \\ &\cup [t_1 + \delta + 2\delta\phi + 2\delta m, t]\end{aligned}$$

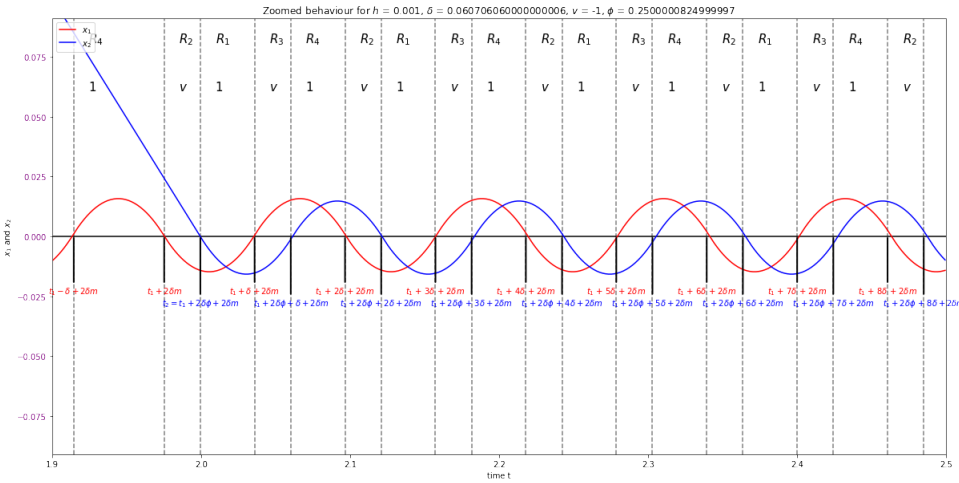


Figure 32: Zoomed in behaviour of $x_1$ and $x_2$ components as $x$ hits $\Sigma_2$ when $\phi \in [0, \frac{1}{2}]$, with the "waves" in phase. At the top is the region $x$ is currently in, and the current velocity in the $z$-component, which is determined by the combination of signs of $x_1$ and $x_2$.

In these intervals, the $z$-component velocity is 1, $v$, 1 respectfully. Therefore (29) becomes

$$\dot{x}_3(t) = \frac{1}{\delta}\left(\int_{t-\delta}^{t_1+\delta+2\delta m} 1ds + \int_{t_1+\delta+2\delta m}^{t_1+\delta+2\delta\phi+2\delta m} vds + \int_{t_1+\delta+2\delta\phi+2\delta m}^{t} 1ds\right)$$

$$\begin{aligned}
&= \frac{1}{\delta}[1(t_1+\delta+2\delta m-(t-\delta)) \\
&\quad +v(t_1+\delta+2\delta\phi+2\delta m-(t_1+\delta+2\delta m) \\
&\quad +1(t-(t_1+\delta+2\delta\phi+2\delta m))] \\
&= \frac{1}{\delta}[1(t_1+2\delta+2\delta m-t) \\
&\quad +v(2\delta\phi) \\
&\quad +1(t-t_1-\delta-2\delta\phi-2\delta m))] \\
&= \frac{1}{\delta}[(\delta-2\delta\phi)+2\delta v\phi] = (1-2\phi)+2v\phi = 1+2\phi(v-1)
\end{aligned}$$

Taking $t$ in any other interval where where $H_1$ or $H_2$ is constant gives the same result. However, such intervals do not have the same form when $\phi \geq \frac{1}{2}$, which may explain the discrepancy of my result and Example 4.3 in [35].
Now assume $\phi \in [\frac{1}{2}, 1]$. See Figure 33.

Say $t \in [t_1+\delta+2\delta\phi+2\delta m, t_1+3\delta+2\delta m]$. Then

$$t-\delta \in [t_1+2\delta\phi+2\delta m, t_1+2\delta+2\delta m]$$

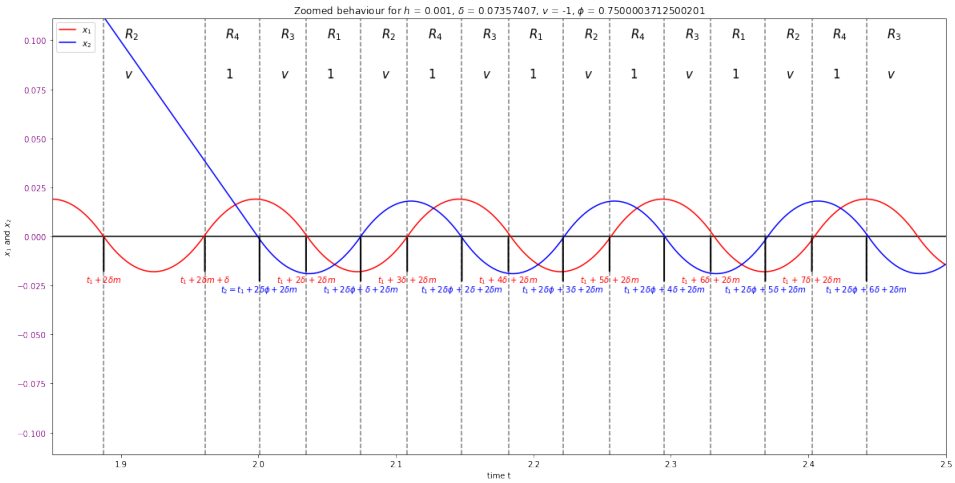


Figure 33: Zoomed in behaviour of $x_1$ and $x_2$ components as $x$ hits $\Sigma_2$ when $\phi \in [\frac{1}{2}, 1]$. Here the "waves" are out of phase. At the top is the region $x$ is currently in, and the current velocity in the $z$-component.

Therefore

$$\begin{aligned}[t-\delta, t] &= [t-\delta, t_1 + 2\delta + 2\delta m] \\ &\cup [t_1 + 2\delta + 2\delta m, t_1 + \delta + 2\delta\phi + 2\delta m] \\ &\cup [t_1 + \delta + 2\delta\phi + 2\delta m, t]\end{aligned}$$

In these intervals, the $z$-component velocity is $v$, 1, $v$ respectfully. Therefore (29) becomes

$$\dot{x}_3(t) = \frac{1}{\delta}\left(\int_{t-\delta}^{t_1+2\delta+2\delta m} v ds + \int_{t_1+2\delta+2\delta m}^{t_1+\delta+2\delta\phi+2\delta m} 1 ds + \int_{t_1+\delta+2\delta\phi+2\delta m}^{t} v ds\right)$$

$$\begin{aligned}&= \frac{1}{\delta}[v(t_1 + 2\delta + 2\delta m - (t-\delta)) \\ &\quad +1(t_1 + \delta + 2\delta\phi + 2\delta m - (t_1 + 2\delta + 2\delta m) \\ &\quad +v(t - (t_1 + \delta + 2\delta\phi + 2\delta m))] \\ &= \frac{1}{\delta}[v(t_1 + 3\delta + 2\delta m - t) \\ &\quad +1(-\delta + 2\delta\phi) \\ &\quad +v(t - t_1 - \delta - 2\delta\phi - 2\delta m))] \\ &= \frac{1}{\delta}[v(2\delta - 2\delta\phi) + 2\delta\phi - \delta] = 2(v - v\phi + \phi) - 1\end{aligned}$$

Similarly, this holds true $\forall t$ and is strikingly independent of $\delta$. In summary, for $t \in [0, a]$ $x(t) = (a - t, b - t, c + t)$. For $t \in [a, b]$ $x_1(t)$ begins oscillating with period $2\delta$ according to (31) while $x_2(t) = b - t$ and $x_3(t) = c + t$. For $t \geq b$,

$x_1(t)$ still oscillates as before, and $x_2(t)$ begins oscillating according to (32) and $x_3(t) = c + \dot{x}_3(t)$, where $\dot{x}_3$ is dependent on $\phi$ and is given in (33).

As note in [35], for $v = -1$, we can make this solution converge to any Filippov sliding trajectory[9]. In this case, the Filippov sliding solution is given by

$$F_s(x) = (0, 0, \lambda'), \quad \lambda' \in [-1, 1]$$

Let

$$\delta_n = \frac{t_2 - t_1}{2n + 1 - \lambda'}$$

which gives $m_{\delta_n} = n$, $\phi_{\delta_n} = \frac{1-\lambda'}{2}$ and so the expression for $x_3^{\delta_n}(t)$ is given by

$$x_3^{\delta_n}(t) = c + \lambda'(t - t_2)(0, 0, 1)$$

for $t \geq t_2$. Therefore we can take $\delta_n$ to 0 while holding $\phi_{\delta_n}$ constant and achieve any of the Filippov sliding trajectories, in agreement with Conjecture 5.4. Below are some plots of the behaviour of the time delay method as applied to this PWS. For this I had to create a discrete version of time delay in order to have give a new vector field, which I explain later.

I wished to verify result (33). See Figure 39 which confirms my result.

## 5.6 Euler Method Example

We shall finish our survey of approximation methods by attempting to apply Euler method to the Example 5.13 that we have now approached from many directions.

$$f_1 = (1, 1, 1),\ f_2 = (1, -1, v),\ f_3 = (-1, 1, v),\ f_4 = (-1, -1, 1) \tag{34}$$

with $v \leq 1$ and the separating surfaces being the $x$ and $y$ plane, i.e. $h_1(x, y, z) = x$, $h_2(x, y, z) = y$, with initial condition $x_0 = (a, b, c) \in R_4$ such that $b > a > 0$, so that we are guaranteed to hit $\Sigma_1$ first.

Let the timestep be $h$, with $h \neq \frac{a}{N}$ or $\frac{b}{N}$ for any $N \in \mathbb{Z}$. We start in $R_1$ so that the difference is given by $x_{n+1} = x_n - h = a - h(n+1)$. Crossing $\Sigma_2$ happens when $x_{n+1} < 0 < x_n \implies a - h(n+1) < 0 < a - hn \implies \frac{a}{h} - 1 < n < \frac{a}{h}$. Therefore we cross $\Sigma_2$ after $n_x = \lfloor \frac{a}{h} \rfloor$. For every step after this (regardless of the $y$-value due to the symmetry of the system), $x$ will be jumping back and forth over $\Sigma_2$ and moreover:

$$x_{n_x+k} > 0 \text{ for } k \text{ even},\ x_{n_x+k} < 0 \text{ for } k \text{ odd}$$

Similarly for $y_n$, we cross $\Sigma_1$ after $n_y = \lfloor \frac{b}{h} \rfloor$ and:

$$y_{n_y+k} > 0 \text{ for } k \text{ even},\ y_{n_y+k} < 0 \text{ for } k \text{ odd}$$

[9] I believe the same is true for general $v$, but was only able to show this numerically

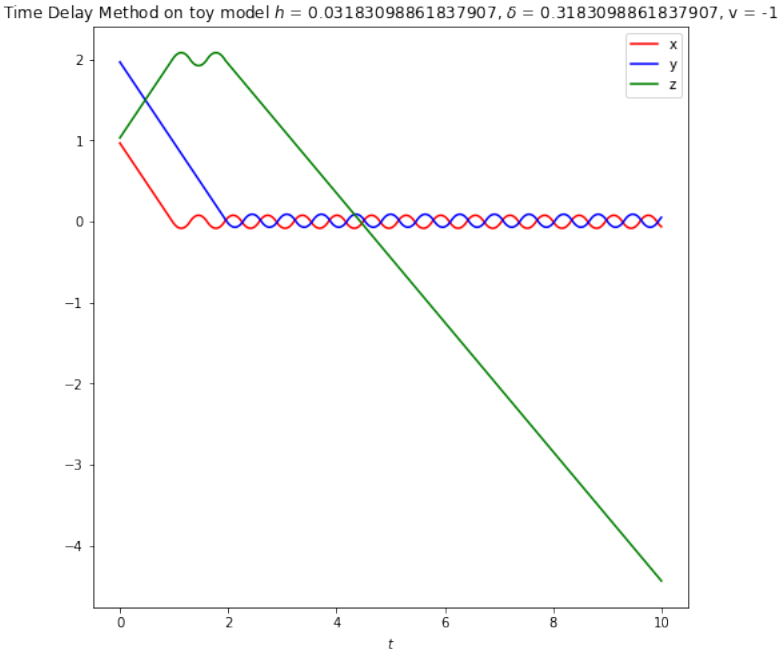


Figure 34: Plot of discrete Time Delay method applied to Example 5.20 with $\delta \approx 0.3$ and $v = -1$. Notice how the $x_1, x_2$ "waves" are very out of phase ($\phi \approx 0.57$), which causes the motion in the $x_1x_2$-plane to behave as in Figure 35. Therefore the orbits spend most of their time in $R_2$ and $R_3$ and hence the gradient of $x_3$ is close to $v = -1$.

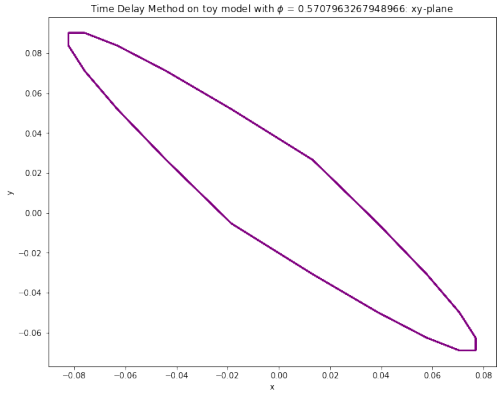


Figure 35: Plot of discrete Time Delay method applied to Example 5.20 with $\delta \approx 0.3$ and $v = -1$ in the $x_1x_2$-plane. Since $\phi \approx 0.57$, this cause the major axis of this ellipse-like shape to be from $R_2$ to $R_3$ and hence the trajectory spends most of it's time under the influence of the vector fields in $R_2$ and $R_3$.

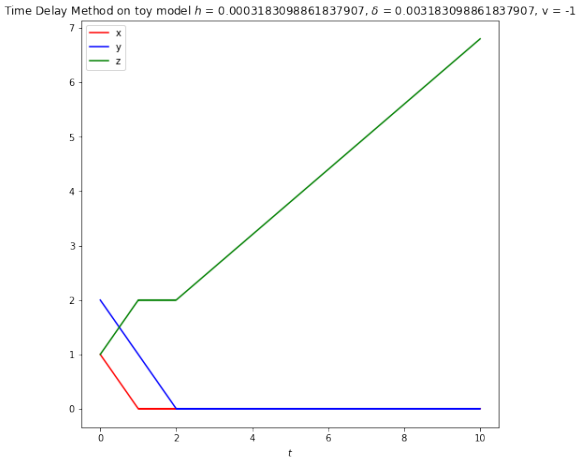


Figure 36: Plot of discrete Time Delay method applied to Example 5.20 with $\delta \approx 0.003$ and $v = -1$. A zoomed in version in Figure 37 shows that the $x_1, x_2$ "waves" are in phase ($\phi \approx 0.07$), which causes the motion in the $x_1x_2$-plane to behave as in Figure 38. Therefore the orbits spend most of their time in $R_1$ and $R_4$ and hence the gradient of $x_3$ is close to 1.

So whether $x_{n_y} = x_{n_x+(n_y-n_x)}$ is greater or less than zero (whether we enter an $h$ neighbourhood of $\Sigma$ from $R_4$ or $R_2$) depends on the parity of $n_y - n_x = \lfloor \frac{b}{h} \rfloor - \lfloor \frac{a}{h} \rfloor$. Let $\sigma = 0$ for $n_y - n_x$ even, $\sigma = 1$ for $n_y - n_x$ odd. Regardless of the parity, we will be stuck in a 2-cycle past $n_y$:

$$\begin{aligned} x_{n_y} &= a - hn_x + h\sigma, & y_{n_y} &= b - hn_y, \\ x_{n_y+1} &= a - (h \pm 1)n_x + h\sigma, & y_{n_y+1} &= b - (h+1)n_y, \\ x_{n_y+2} &= a - (h \pm 1)n_x + h\sigma \mp h = x_{n_y}, & y_{n_y+2} &= b - (h+1)n_y - h = y_{n_y} \end{aligned}$$

Note that this 2-cycle will be between $R_4$ and $R_1$ (resp. $R_2$ and $R_3$) for $n_y - n_x$ even (resp. odd).
During all this time, $z$ will also being changing:

$$z_{n_x} = c + hn_x, z_{n_x+1} = c + hn_x + hv, z_{n_x+2} = c + hn_x + hv + h$$

and in general, for $l \leq n_y$:

$$z_{n_x+l} = c + hn_x + h\left(\left\lfloor \frac{l+1}{2} \right\rfloor v + \left\lfloor \frac{l}{2} \right\rfloor\right)$$

So the parity of $n_y - n_x$ will determine the value of $z_{n_y}$, just before we enter the two cycle. If $n_y - n_x$ is even, $z_{n_y} = c + hn_x + h\left(\frac{n_y-n_x}{2}\right)(v+1)$, and in the next time step enters the 2-cycle between $R_4$ and $R_1$ which both contribute 1

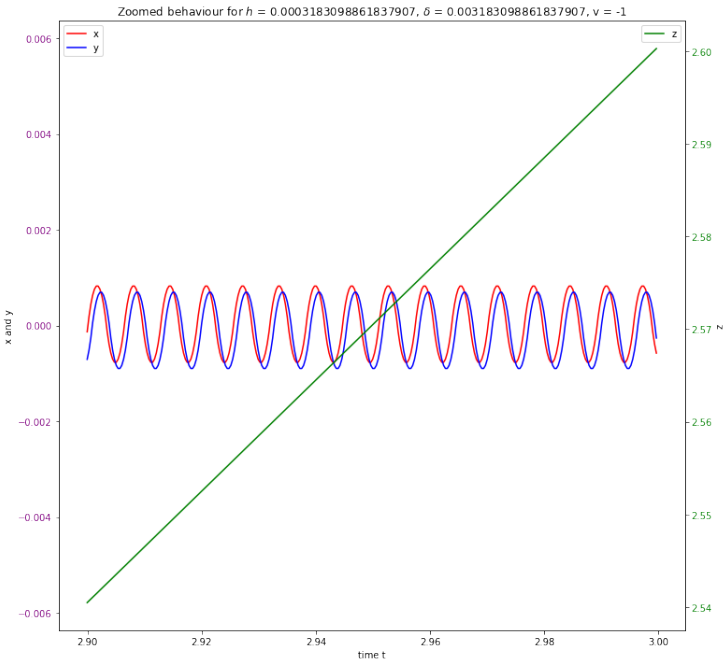


Figure 37: A zoomed in version of Figure 36, showing how the $x_1, x_2$ oscillates over $\Sigma_1, \Sigma_2$ respectfully in phase with each other, causing the gradient of $x_3$ to be closer to 1.

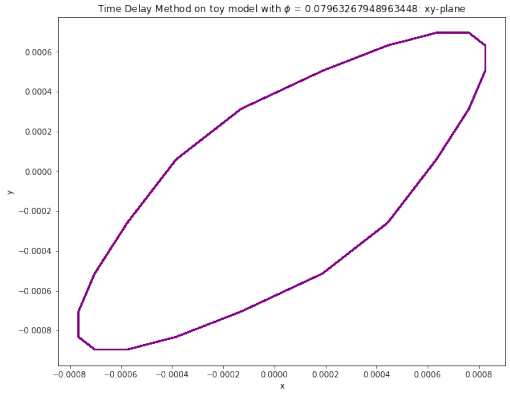


Figure 38: Plot of discrete Time Delay method applied to Example 5.20 with $\delta \approx 0.003$ and $v = -1$ in the $x_1x_2$-plane. Since $\phi \approx 0.07$, this cause the major axis of this ellipse-like shape to be from $R_1$ to $R_4$ and hence the trajectory spends most of it's time under the influence of the vector fields in $R_1$ and $R_4$.

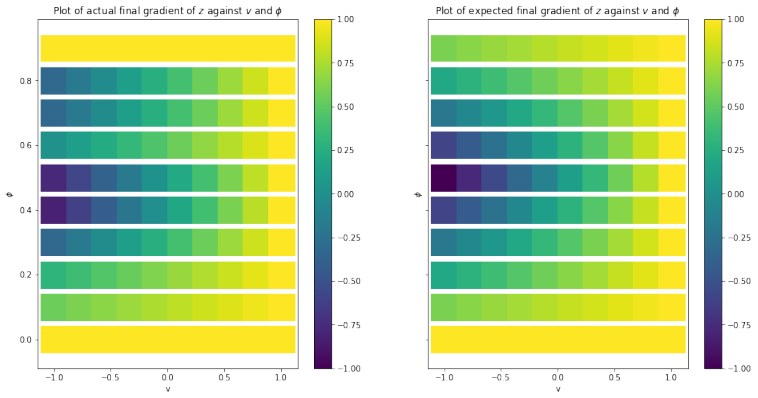


Figure 39: On the left: a color plot showing the discrete Time Delay method applied to 5.20 for various values of various values of $0 \leq \phi < 1$ and $-1 \leq v \leq 1$. The color represents the overall gradient of the $x_3$ component after time $t_2 = c$. On the right: a color plot showing the values of (33) for various values of $0 \leq \phi < 1$ and $-1 \leq v \leq 1$. As we can see, the colormaps are similar which confirms the veracity of 33. They become even more similar for smaller time steps, but I lost the image showing this and cannot replicate it due to large computation times for small time step and the number of different simulations ran.

to the velocity in the $z$-direction i.e.:

$$z_{n_y+m} = c + hn_x + h\left(\frac{n_y - n_x}{2}\right)(v+1) + mh$$

so that $z$ travels with velocity 1 after entering a $h$-neighbourhood of $\Sigma$. On the other hand if $n_y - n_x$ is odd, $z_{n_y} = c + hn_x + h\left(\frac{n_y - n_x - 1}{2}\right)(v+1) + hv$, and in the next time step enters the 2-cycle between $R_2$ and $R_4$ which both contribute $v$ to the velocity in the $z$-direction i.e.:

$$z_{n_y+m} = c + hn_x + h\left(\frac{n_y - n_x - 1}{2}\right)(v+1) + mhv$$

so that $z$ travels with velocity $v$ after entering a $h$-neighbourhood of $\Sigma$.

So in the case of this system, the Euler method only selects the Filippov sliding solutions with extreme gradients $v$ and 1. One could easily generalise this system to higher dimensions in order to obtain a lower bound on the number of distinct Euler polygonal limits for a given system. This is in agreement with our numerical simulations. Below are graphs of simulations according to to the Euler method with varying parity of $n_y - n_x$.

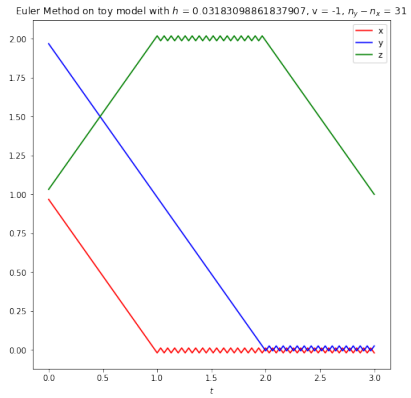


Figure 40: Plot of Euler method applied to 5.13 with odd $n_y - n_x$.

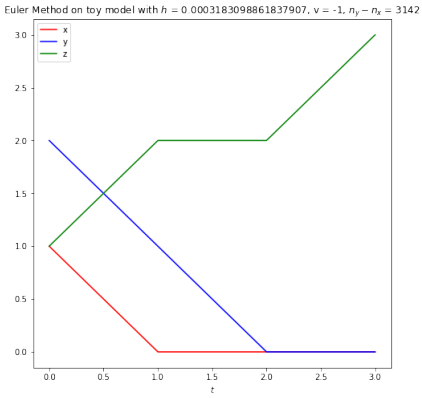


Figure 41: Plot of Euler method applied to 5.13 with odd $n_y - n_x$.

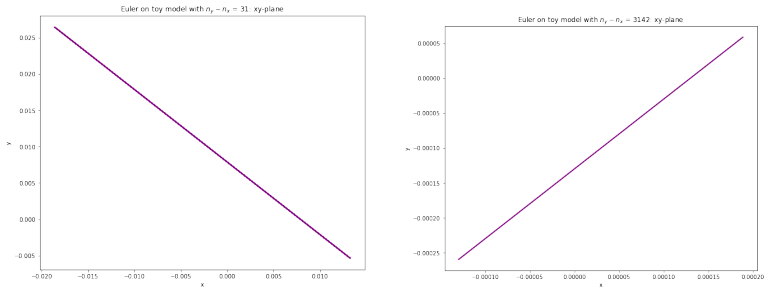


Figure 42: On the left: Plot in $xy$-plane of Euler method applied to 5.13 with low time step and odd $n_y - n_x$. On the right: The same but with a smaller time step and even $n_y - n_x$. As we can see, the size of the timestep does not matter as we are stuck in a 2-cycle no matter.

# 6 Application

We are now armed with 2 powerful solution concepts in the bilinear and time delay methods. Recall our goal was to find an analytically "predictable" solution concept to compare with the usually unpredictable Euler solution concept. In this section, we have numerical simulations of each method as applied to the original problem of $\epsilon$-IQL system as applied to the Prisoner's Dilemma (The PWS given in Figure 1).

## 6.1 Numerical Simulations of Different Methods

What follows are the discrete implementations of the different solutions concepts I used to create the numerical simulations. In all of them, we compute a vector $f$ according to the solution concept, and then take a timestep in the direction of that vector with the Euler method.

### 6.1.1 Discrete Euler Method

This is the simplest to implement given it is already a discrete method. Given an initial value $x_0$ and a timestep $h$, we define $x_n$ as:

$$x_n = x_{n-1} + hf(x_{n-1})$$

where $f(x) = f_i(x)$ for $x \in R_i$. If $h_k(x) = 0$, then we assume that $h_k(x) > 0$ so that we always get a vector field that will determine the next value. In practice however, $h_k(x)$ was never zero for any of the simulations we ran for this.

### 6.1.2 Discrete Bilinear Method

I was not successful in creating a discrete approximation of the Bilinear method, but I will outline the approach I attempted below.
We define $\Sigma^r \coloneqq \{x \in \mathbb{R}^n : \left|h_1(x)\right| < r \text{ and } \left|h_2(x)\right| < r\}$, which will be a square neighbourhood around $\Sigma$. Outside of $\Sigma^r$, we follow the Euler method from before[10]. Within $\Sigma^r$, we check first Assumptions 5.17 are satisfied. If they are not, we do not have the conditions for the bilinear method to be applied, and we follow the Euler method from before. If they are satisfied, we set

$$f(x) \coloneqq F_B(x) = (1-\bar{\alpha})(1-\bar{\beta})f_1 + (1-\bar{\alpha})\bar{\beta}f_2 + \bar{\alpha}(1-\bar{\beta})f_3 + \bar{\alpha}\bar{\beta}f_4,\ \bar{\alpha}, \bar{\beta} \in [0,1],$$

where $(\bar{\alpha}, \bar{\beta})$ are roots of $P$ and $Q$ respectively, as given in (16, 18) and $g(\bar{\beta}) = \bar{\alpha}$ and $h(\bar{\alpha}) = \bar{\beta}$, as given in (19,17).
In my code, I found the roots of the polynomials $P$ and $Q$ using both the a symbolic solver from Sympy and a Newton Raphson method from Scipy. From there we restrict to those roots such that $h$ applied to the roots of $Q$ and $g$ applied to the roots of $P$ are both in $[0,1]$. If no such roots exist (or there are

[10]Note if $r = 0$, we are saying that there is no fuzzy region and so we follow the Euler method everywhere.

more than one pair), we default to the Euler method i.e. $f(x) = f_i(x)$, $x \in R_i$. If such a pair of roots do exist, they are the unique $(\bar{\alpha}, \bar{\beta})$ described in Subsection 5.4 that solve the system of equations (15).
However, my code was unable to handle any complicated PWSs: it matched the behaviour of Example 5.18, but would consistently give more than one valid $(\bar{\alpha}, \bar{\beta})$ pair when applied to $\epsilon$-IQL for the Prisoner's Dilemma and give very different behaviour from the Euler method. I believe this is due to exit conditions in which the inequalities in Assumptions 5.17 become equalities, which causes the solutions to 15 to be non-unique. In my simulations, $r$ was set as 5 times $h$.

### 6.1.3 Discrete Time Delay Method

This was very complicated to construct, but did successfully predict the behaviour of Example 5.20 as seen in Figure 36, and gave similar behaviour to the Euler $\epsilon$-IQL as applied to Prisoner's Dilemma. The method goes as follows:
$D_{n-1}$ is defined as $t_{n-1} - \delta$ rounded up to the closest multiple of $h$ so that $D_{n-1} = t_m$ for some $m$. Then there will be $N$ multiples of $h$ between $D_{n-1} = t_m$ and $t_{n-1}$. We define $f(x_{n-1}) = \frac{1}{n-m-1} \sum_{j=m}^{n-1} f_i(x_j)$ which is the average of the vector fields visited from $D_{n-1} = t_m$ to $t_{n-1}$. We then use the Euler method to define the next iterate $x_n$ as before. Note that this requires $\delta$ to be quite a bit larger than $h$ so that we have a high number of samples to average over, although this does cause computation time to increase. In my simulations, $\delta$ was usually set as 10 times $h$.

### 6.1.4 Discrete Results of $\epsilon$-IQL

In what follows we show the behaviour of the Discrete Time Delay method applied to $\epsilon$-IQL for the Prisoner's Dilemma. As we can see, the behaviour is remarkably similar to the behaviour seen in Section 4.1.

# 7 Conclusion

In this report I aimed to find whether the Euler $\epsilon$-IQL as applied to the Prisoner's Dilemma was chaotic. I quickly found that the situation was much more complex: due to the Realisability Theorem (Theorem 5.14), we know that it will converge to some Filippov sliding trajectory when the indifference lines are attractive, although it is not clear which Filippov sliding solution this will be. Indeed there seems to be a gap in the literature for a detailed analysis of the behaviour of PWSs under the Euler method solution concept, despite its popularity as a numerical implementation.

It is perhaps clearer, as in Section 5.6, that if the Filippov solution for the PWS generated by Euler $\epsilon$-IQL as applied to the Prisoner's Dilemma is non-unique, we can descend to different solution trajectories by altering which

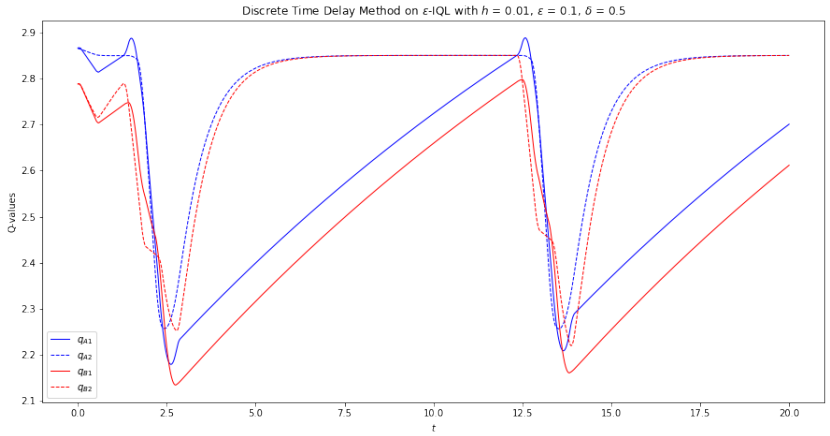


Figure 43: Discrete Time Delay method applied to $\epsilon$-IQL for the Prisoner's Dilemma, with $\delta = 0.5$, $h = 0.01$, $\epsilon = 0.1$. This is a very smooth approximation that seems to match much of the behaviour described in section 5.2.

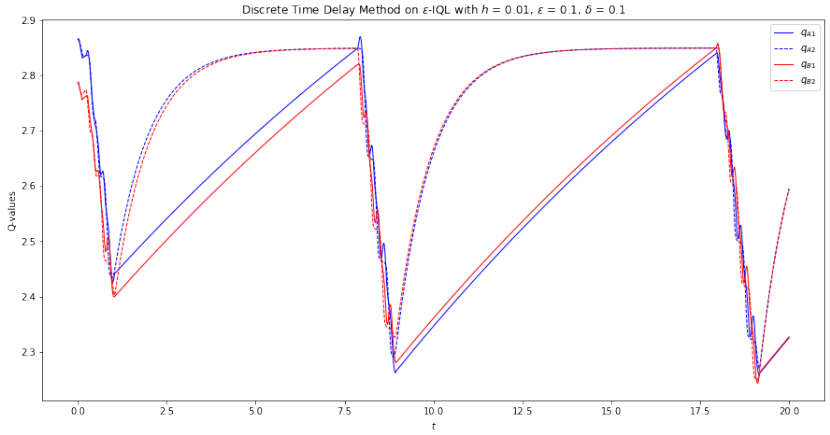


Figure 44: Discrete Time Delay method applied to $\epsilon$-IQL for the Prisoner's Dilemma, with $\delta = 0.1$, $h = 0.01$, $\epsilon = 0.1$. It closely emulates the behaviour of the Euler method, but with the sliding in the Opportunistic Defecction phase being smooth.

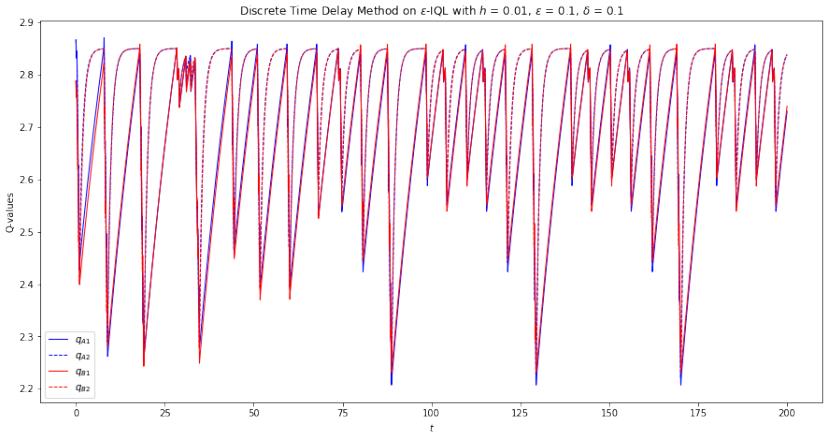


Figure 45: Discrete Time Delay method applied to $\epsilon$-IQL for the Prisoner's Dilemma, with $\delta = 0.1$, $h = 0.01$, $\epsilon = 0.1$ for 200 time units. No apparent transient behaviour.

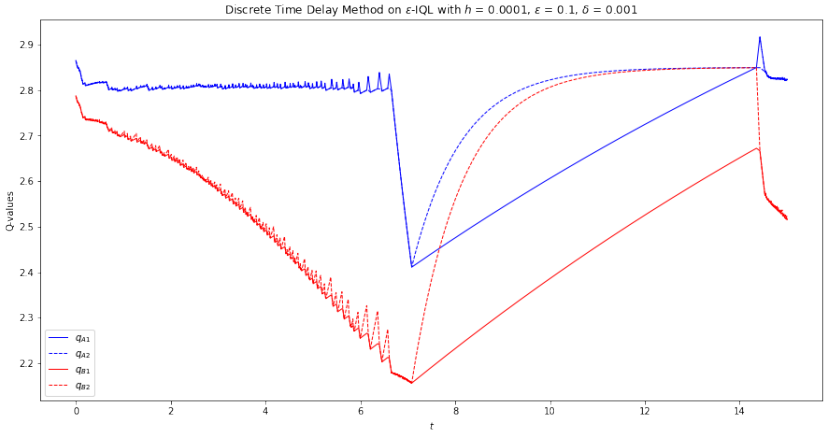


Figure 46: Discrete Time Delay method applied to $\epsilon$-IQL for the Prisoner's Dilemma, with $\delta = 0.001$, $h = 0.0001$, $\epsilon = 0.1$. Althought the length of the cycle is slightly shorter, notice how similar this is to the Euler method in Figure 14.

sequence $h_n$ of time steps we let go to 0. Some of these solution trajectories should be transient and some should not be, as discussed in Section 4.1. Whether the ones that are non-transient are chaotic or merely periodic is still unclear. Perhaps bounds on the distance between Euler polygons and the set of Filippov sliding trajectories as given in [31, 32] would help to at least give some insight into how the Euler polygons act as $h \to 0$. A derivation of the set of Filippov solutions of this PWS would be tedious but not difficult. If the set was singleton, then we could reason that this is the solution to Euler $\epsilon$-IQL as the time step goes to 0, and either the transient or non-transient are just special cases. However, I believe that it will be multi-valued and both are valid Filippov trajectories but with different values of the parameter in the one-parameter family of Filippov solutions. I did not tackle the problem of exit conditions in this report, which seem fundamental to the apparent chaotic behaviour coming from the Opportunistic Defection phase.

I was able to survey some alternative solution concepts that could be applied to $\epsilon$-IQL for Prisoner's Dilemma. Due to the fact that Bilinear method only selects a single solution, it is unlikely this is the solution to which the Euler method converges. The Time Delay method however, is a lot more powerful, if Conjecture 5.4 is to be believed. Perhaps, as in Example 5.20, it is possible to find a trajectory which can converge to any of the Filippov trajectories of the $\epsilon$-IQL for Prisoner's Dilemma PWS. From Section 6.1.4, we saw that the Discrete Time Delay method closely resembled the behaviour of the Euler method. From here, it may be possible to find which specific values of $\delta$ give transient or chaotic behaviour and draw some correspondence between these and the values of the time step $h$ in the Euler method which give transience or chaos.

# References


[1] Watkins, c. (1989) Learning From Delayed Rewards `https://www.cs.rhul.ac.uk/ chrisw/thesis.html`

[2] Bellman, R. (1957). A Markovian Decision Process. Journal of Mathematics and Mechanics. 6 (5): 679–684. JSTOR 24900506. `http://www.iumj.indiana.edu/IUMJ/FULLTEXT/1957/6/56038`

[3] Watkins, C.J.C.H., Dayan, P. (1992) Q-learning. Mach Learn 8, 279–292 . `https://doi.org/10.1007/BF00992698`

[4] Francisco S. Melo, Convergence of Q-learning: a simple proof. `http://users.isr.ist.utl.pt/ mtjspaan/readingGroup/ProofQlearning.pdf`

[5] S. Nouyan, R. Gross, M. Bonani, F. Mondada and M. Dorigo, "Teamwork in Self-Organized Robot Colonies," in IEEE Transactions on Evolutionary Computation, vol. 13, no. 4, pp. 695-711, Aug. (2009).

[6] Bruce Bueno de Mesquita. Game theory, political economy, and the evolving study of war and peace. American Political Science Review, 100(4):637–642, November (2006).

[7] Shlomit Hon-Snir, Dov Monderer, and Aner Sela. A learning approach to auctions. Journal of Economic Theory, 82:65–88, November (1998).

[8] K. Tuyls, K. Verbeeck, and T. Lenaerts. (2003) A selection mutation model for Q-learning in multi-agent systems. AAMAS-2003, pages 693–700 https://doi.org/10.1145/860575.860687

[9] Zawadzki, E., Lipson, A., and Leyton-Brown, K. (2008) Empirically evaluating multiagent learning algorithms. https://arxiv.org/abs/1401.8074

[10] Galstyan, A., Czajkowski, K., & Lerman, K. (2004) Resource allocation in the grid using reinforcement learning. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04) (pp. 1314–1315). Washington, DC, USA: IEEE Computer Society.

[11] M. Wunder, M. Littman, and M. Babes. (2010) Classes of multiagent Q-learning dynamics with $\epsilon$-greedy exploration. In Proc. of ICML-2010,

[12] E. Gomes and R. Kowalczyk. (2009) Dynamic analysis of multiagent Q-learning with $\epsilon$-greedy exploration. In Proc. of ICML-2009.

[13] M. Falcone, (2005) Numerical Methods for Differential Games based on Partial Differential Equations. https://doi.org/10.1142/S0219198906000886

[14] S. van Strien and B. Winckler, (Working Paper) On the Dynamics of the $\epsilon$-Greedy Q-learning Algorithm

[15] S. van Strien and B. Winckler, (Working Paper) Learning in Multi Agent Games

[16] Lind, D. & Marcus, B., 1995. Introduction to Symbolic Dynamics and Coding. Cambridge University Press, Cambridge, U.K.

[17] Casey, R., Jong, H. & Gouzé, JL (2006). Piecewise-linear Models of Genetic Regulatory Networks: Equilibria and their Stability. J. Math. Biol. 52, 27–56 https://doi.org/10.1007/s00285-005-0338-2

[18] I, Utkin & Chang, Hao-Chi. (2002). Sliding mode control on electro-mechanical systems. Mathematical Problems in Engineering. 8. 10.1080/10241230306724.

[19] Leine, R. I. (2000). Bifurcations in discontinuous mechanical systems of the Fillippov-type. Technische Universiteit Eindhoven. https://doi.org/10.6100/IR533239

[20] Walker, S.V., Leine, R.I. (2019). Set-valued anisotropic dry friction laws: formulation, experimental verification and instability phenomenon. Nonlinear Dyn 96, 885–920. https://doi.org/10.1007/s11071-019-04829-6

[21] Glocker, Ch. (2001). Set-Valued Force Laws: Dynamics of Non Smooth Systems. Lecture Notes in Applied Mechanics, vol. 1. Springer, Berlin, section 1.1: Friction Laws

[22] Aubin, Jean Pierre and Cellina, A., (1984) Differential Inclusions: Set-Valued Maps and Viability Theory, Springer-Verlag https://doi.org/10.1002/zamm.19870670206

[23] A. Bressan, Singularities of Stabilizing Feedbacks, Rend. Sem. Mat. Univ. Pol. Torino Vol. 56, 4 (1998). http://directory.umm.ac.id/Journals/Journal_of_mathematics/OTHER/87.pdf

[24] Bernardo, M., Budd, C., Champneys, A.R., Kowalczyk, P., (2008) Piecewise-smooth Dynamical Systems Theory and Applications, Springer-Verlag London, https://doi.org/10.1007/978-1-84628-708-4

[25] L. Dieci, C. Elia, (2015). Piecewise smooth systems near a Co-dimension 2 discontinuity manifold: Can one say what should happen? https://arxiv.org/abs/1508.02639

[26] V. I. Utkin, (1992) Sliding Modes in Control and Optimization https://doi.org/10.1007/978-3-642-84379-2

[27] L. Dieci, C. Elia, L. Lopez (2012). A Filippov sliding vector field on an attracting co-dimension 2 discontinuity surface, and a limited loss-of-attractivity analysis. https://doi.org/10.1016/j.jde.2012.11.007

[28] J.C. Alexander, T. Seidman (1998), Sliding modes in intersecting switching surfaces, I: Blending, Houston J. Math. 24 545–569.

[29] A.F. Filippov, (1988), Differential Equations with Discontinuous Righthand Sides

[30] T. Ważewski, (1962) Sur une généralisation de la notion des solutions d'une équation au contingent. (French) Bulletin de l'Académie Polonaise des Sciences. Série des Sciences Mathématiques, Astronomiques et Physiques , 1962, Vol.10, p.11-15. MR0153944

[31] Dontchev, A.L., Farkhi, E.M. Error estimates for discretized differential inclusions. Computing 41, 349–358 (1989). https://doi.org/10.1007/BF02241223

[32] A. Dontchev, F. Lempio (1992) Difference Methods for Differential Inclusions: A Survey, SIAM Review Vol. 34, No. 2, pp. 263-294, June 1992 https://doi.org/10.1137/1034050

[33] J.C. Alexander, T. Seidman (1998), Sliding modes in intersecting switching surfaces, II: Hysteresis, Houston J. Math. 25

[34] L. Dieci, F. Difonzo, (2017) The Moments Sliding Vector Field on the Intersection of Two Manifolds http://dx.doi.org/10.1007/s10884-015-9439-9

[35] Giorgio Fusco, Nicola Guglielmi (2011). A regularization for discontinuous differential equations with application to state-dependent delay differential equations of neutral type, Journal of Differential Equations, Volume 250, Issue 7, Pages 3230-3279, https://doi.org/10.1016/j.jde.2010.12.013.