# Bifurcations In the Geometry of the Attractor In Border Collision Normal Form

Xitian Huang
Student ID: 10432171

*Supervisor: Paul Glendinning*

May 2022

# Contents

**1 Introduction** **1**
- 1.1 Preliminaries . . . 1
  - 1.1.1 Border Collision Normal Form . . . 1
  - 1.1.2 Absorbing Area and Attractor . . . 2
- 1.2 Construction of the Attractor . . . 2
  - 1.2.1 Conditions for the Attractors . . . 2
  - 1.2.2 Geometric Behaviour . . . 4
- 1.3 Programming . . . 5

**2 Bifurcations** **7**
- 2.1 Bifurcation On the Border . . . 7
  - 2.1.1 Theoretical Setup . . . 7
  - 2.1.2 $P_2$ on the Right . . . 9
  - 2.1.3 $P_2$ on the Left . . . 10
  - 2.1.4 Method of Critical Curves . . . 10
- 2.2 Bifurcation On the Boundary . . . 12
  - 2.2.1 When $P_2$ is On the Right . . . 12
  - 2.2.2 When $P_2$ is On the Left . . . 15
- 2.3 More Bifurcations . . . 16

**3 Conclusion** **19**

**Appendix** **22**

# Chapter 1

# Introduction

The study of piecewise smooth systems with a low dimensional invariant submanifold has brought the attention of researchers in the field of nonlinear mathematics in the past decades. With the advancement in computer science, many numerical analyses were accomplished to further aid the theoretical studies of such dynamical systems. In particular, the bifurcations in such systems have many interesting behaviours, including the generation of absorbing regions, attractors, periodic orbits, etc., as we will demonstrate later.

This project is devoted to studying the attractors of what is known as *border collision normal form*, transformed as a first-order approximation to any two dimensional piecewise smooth map. Specifically, we will look at the bifurcations in the geometry of attractors in its parameter space.

## 1.1 Preliminaries

### 1.1.1 Border Collision Normal Form

Consider a two dimensional piecewise smooth map $f(x, y; \rho)$ that has dependence on a parameter $\rho$ of the form:

$$f(x, y; \rho) = \begin{cases} f_1(x, y; \rho) & \text{for } x, y \in R_a \\ f_2(x, y; \rho) & \text{for } x, y \in R_b. \end{cases} \tag{1.1}$$

$f_1$ and $f_2$ are assumed to be continuously differentiable in their corresponding domains $R_a$ and $R_b$. Notice that at the *border* $\Gamma_\rho$ that separates $R_a$ and $R_b$, the piecewise map $f$ formed by $f_1$ and $f_2$ is continuous, but its derivative is not continuous. The one-sided derivatives at the border are assumed to be finite.

Illustrated by Nusse and Yorke (1992), such a piecewise smooth map can be transformed by a change of coordinates, up to first-order affine approximation, to a *border collision normal form*:

$$F(x,y;\mu) = \begin{cases} \begin{pmatrix} \tau_L & 1 \\ -\delta_L & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \mu \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \text{for } x \leqslant 0 \\ \begin{pmatrix} \tau_R & 1 \\ -\delta_R & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \mu \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \text{for } x > 0, \end{cases} \tag{1.2}$$

where the new border is the $y$-axis: $x = 0$, which divides the space into two regions: $L$ for left, $R$ for right. The new parameter $\mu$ is obtained by rescaling $\rho$. $\tau$ and $\delta$ are the trace and determinant of the Jacobian matrix of both maps $f$ and $F$ on either side. This is because the trace and determinant are invariant under the transformation of coordinates.

### 1.1.2 Absorbing Area and Attractor

**Absorbing area**
Given a continuous map $T : \mathbb{R}^2 \to \mathbb{R}^2$, a closed connected area $\mathcal{A}$ is an *absorbing area* if $T(\mathcal{A}) \subseteq \mathcal{A}$. When $T(\mathcal{A}) = \mathcal{A}$, $\mathcal{A}$ is called an *invariant absorbing area.*

**Attractor**
An invariant absorbing area $\mathcal{A}$ becomes an *attractor* if there exists an open set $\mathcal{U}$ that contains $\mathcal{A}$ such that $\lim\limits_{n\to\infty} T^n(\mathcal{U}) = \mathcal{A}$ and the orbits of $T$ are dense in $\mathcal{A}$ ($T^n$ denotes the $n$-th iterate of $T$).

Therefore, attractor implies (invariant) absorbing area but the converse is not necessary.

## 1.2 Construction of the Attractor

It is shown by Glendinning and Wong (2011) that there exists an attractor in the form of a finite *Markov partition.* Such attractor has a simple geometry, it is formed by straight lines and together they create a polygon. We will develop the theory based on this and study the bifurcation in its geometry later.

### 1.2.1 Conditions for the Attractors

Consider the normal form in Eq.(1.2). It is a piecewise linear map and hence the magnitude of the parameter $\mu$ does not influence observing the bifurcations in our case, only the sign. When $\mu = 0$, the fixed points on two sides collide and annihilate, this is the standard saddle-node bifurcation and does not concern the study of this project. Therefore, without loss of generality and for convenience, we can set $\mu = 1$.

It is obvious that the map always sends the origin $O(0,0)$ to $P_1(\mu, 0)$. Figure 1.1 gives an illustrative picture. Require the first $n$-th iterate of the map to be on the $y$-axis, $F^n(O) = P_n = (0, y_0)$. This determines the parameters $\tau_R$ and $\delta_R$ if the number of iterations $n$ is specified.

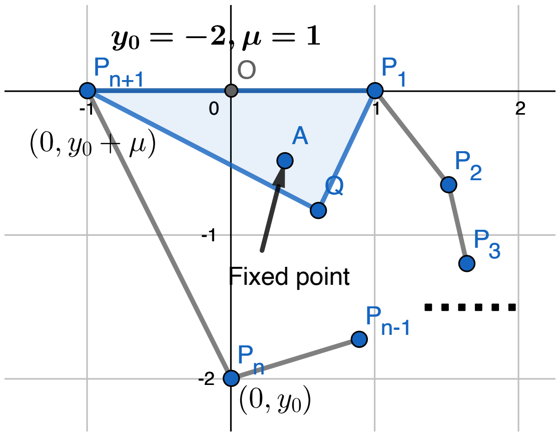


Figure 1.1: Iterated points from $O$ under the map $F$, $O \to P_1 \to P_2 \to ... \to P_n \to P_{n+1} \to Q$, forms a closed convex polygon $P_1P_2...P_nP_{n+1}$ with $n+1$ sides. $y_0 = -2, \mu = 1$.

Then $F(P_n) = P_{n+1} = (y_0 + \mu, 0)$. We need $y_0 + \mu < 0$ so that $P_{n+1}$ is on the left. The fixed point $A$ of $F$ on the right, given by $F(A) = A$ when $x > 0$ is

$$A = \frac{\mu}{1 - \tau_R + \delta_R} (1, -\delta_R) . \tag{1.3}$$

Let point $Q = F(P_{n+1})$ such that the fixed point on the right $A$ is inside the triangle $P_1P_{n+1}Q$ and $Q$ is in the polygon $P_1P_2...P_nP_{n+1}$. If $Q$ is outside this polygon, the iterated points will eventually escape from the polygon and this doesn't form an absorbing area. Since $P_{n+1}$ is on the left, this determines the parameters $\tau_L$ and $\delta_L$ if the coordinate of $Q$ is given.

The reason for the condition that $A$ in $P_1P_{n+1}Q$ is as follows. We look at the area $P_1P_{n+1}Q$, the image of the left area $OP_nP_{n+1}$ under the map $F$ when $x < 0$. If this triangle doesn't cover the fixed point $A$, then the absorbing region may have a "hole" near $A$ or other periodic structures, and the geometry of the attractors becomes much more complicated, as shown in Figure 1.2. This is because the map creates unstable periodic orbits and if $A$ is not in $P_1P_{n+1}Q$, iterated points starting near $A$ will spiral away from it, hence the creation of the hole.

To determine the local stability of the periodic orbits, (Di Bernardo et al.; 2008) one needs to find the modulus of the eigenvalues $\lambda_\pm$ of the square matrix in Eq.(1.2):

$$\lambda_\pm = \frac{\tau \pm \sqrt{\tau^2 - 4\delta}}{2}. \tag{1.4}$$

If both $|\lambda_+|, |\lambda_-| > 1$, the periodic orbits are unstable, as is what happens here.

In summary, we have the following assumptions:

- $\mu > 0$ in Eq.(1.2),
- $F^n(O) = P_n = (0, y_0)$, where $y_0 < -\mu$,

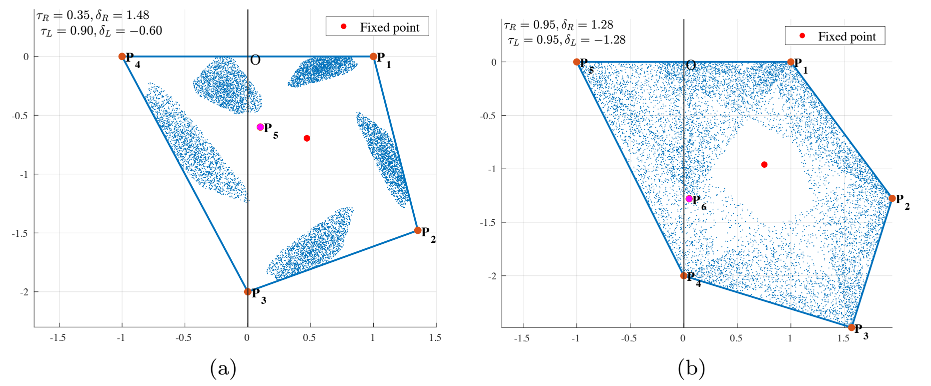


Figure 1.2: Generations of the hole or periodic structure when $P_1P_4P_5$ or $P_1P_5P_6$ doesn't cover the fixed point. 10000 points were plotted in each figure.

- All points $P_1P_2...P_{n-1}$ are on the right,
- Fixed point $A$ is inside the triangle $P_1P_{n+1}Q$,
- $Q$ is in $P_1P_2...P_nP_{n+1}$.

With these conditions, the four parameters $\tau_L, \delta_L, \tau_R, \delta_R$ are determined (by solving linear equations with MATLAB in this project) and $P_1P_2...P_nP_{n+1}$ forms an invariant absorbing area, moreover, an attractor. The proof of a quadrilateral attractor is given in (Glendinning and Wong; 2011).

As mentioned before, the size of $\mu$ does not matter, this also applies to $y_0$. We set $\mu = 1, y_0 = -2$ and will use these values hereafter unless otherwise stated. Figure 1.1 depicted the situation in more detail.

### 1.2.2 Geometric Behaviour

With the above theoretical setup, Figure 1.3 gives 6 figures of the polygonal attractors with different number of sides.

Instead of looking at the set of single points the map sends to, one should look at the iterated lines and areas. If a line $AB$ is mapped to $BC$, then any point on $AB$ will have its image on $BC$. The same applies to areas. This would be particularly useful later when we try to find the lines that form the boundary of the attractor, and the areas that determine the existence of a simple attractor. For instance, when finding the conditions for the attractors in the last section, one requires the fixed point $A$ to be inside the area formed by the triangle $P_1P_{n+1}Q$.

Looking at the plots in Figure 1.3, we see some regions are more "dense" than other regions. This is a "folding action", as described in (Banerjee et al.; 1998). In Figure 1.3(a), $P_1P_3P_4$ is darker. We also see some vague lines inside the polygons, e.g., $P_3P_4$ is the image of $P_2P_3$. The creation of other lines is analogous.

## 1.3 Programming

Help from the computer plays a vital role in this project as we usually need to plot around 40000 points, which means 40000 iterations of the map. In the Appendix, four MATLAB scripts used throughout are included. All Figures that appeared (except the schematic diagrams) in this paper were generated by these codes.

Two of them are defined functions: `f.m` and `InterX.m`.

`f.m` is simply the function in Eq.(1.2). Input a point gives a new point under the map. Input a line (two points) gives a new line by the map (or two lines if it crosses the $y$-axis).

`InterX.m` gives the intersection of two lines.

These two functions need not be altered.

`main.m` is the main script that plots the iteration points. Note that some initial variables need to be modified to adapt to different circumstances.

`label.m` labels the plot. This includes, annotate the points, find the vertices that construct the boundary of the attractor and draw the boundary of the attractor. In this code, depending on the situation, some parts of it need to be commented in order for the code to run. Many of its variables need to be modified manually, e.g., `polygon, pt_start, X_1, X_2, X_3` according to the resulted plot from `main.m`.

Always start from `main.m` and run `label.m` next (modify `label.m` if necessary).

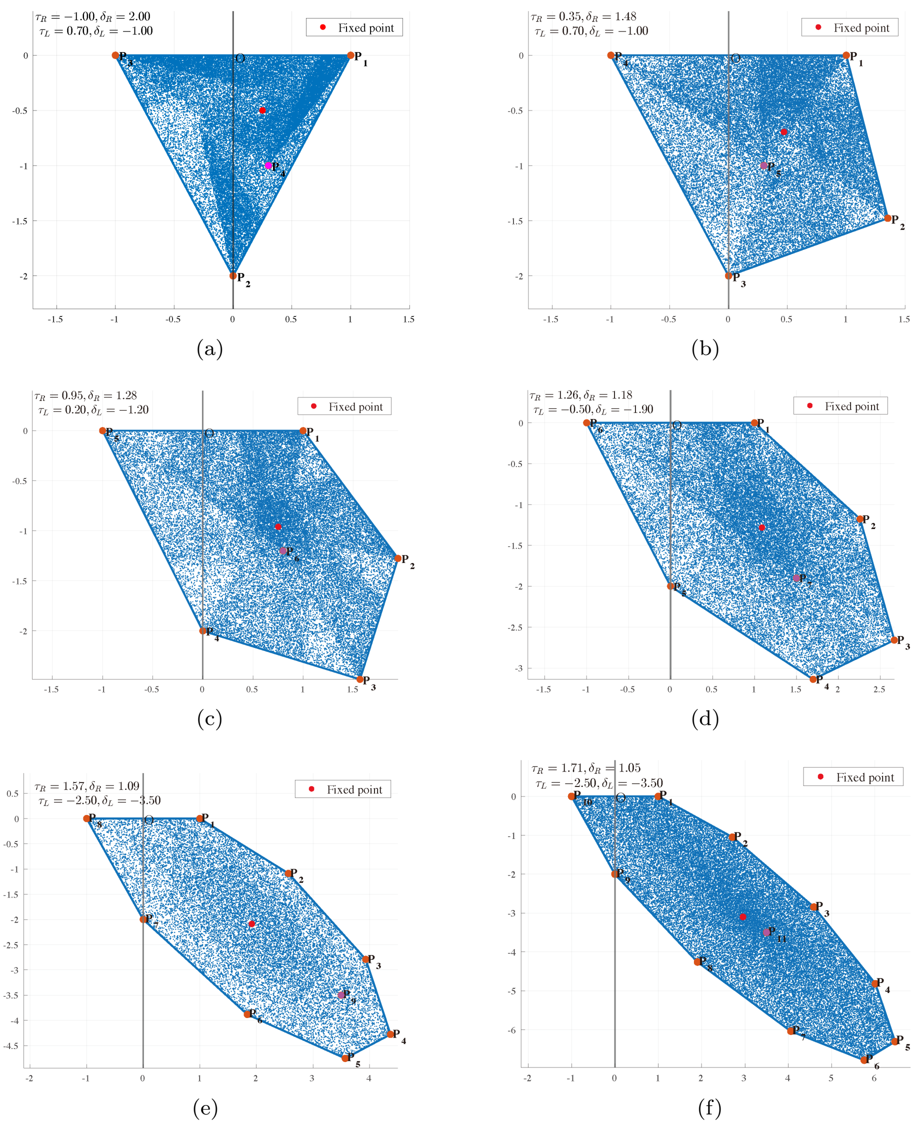


Figure 1.3: The generated simple polygonal attractors with different numbers of sides. The values of parameters are given in each case. 40000 points were plotted in each figure.

# Chapter 2

# Bifurcations

The border collision normal form is a piecewise linear and everywhere continuous map on $\mathbb{R}^2$. Therefore, the perturbation in a small neighbourhood of its parameter space should not impact drastically the behaviour of the system. In what follows, we will study how such perturbation changes the geometry of the previous simple polygonal attractors.

## 2.1 Bifurcation On the Border

Let's start from the simplest case in Figure 1.3(a) where we have a triangular attractor. We will move point $P_2$ by a small amount in the following way.

### 2.1.1 Theoretical Setup

Suppose the coordinate vector of $P_2$ is $\boldsymbol{x_2}$, it is the second iterate from the origin of the map $\boldsymbol{F}$ in Eq.(1.2), i.e.

$$\boldsymbol{F}^2(\mathbf{0}) = \boldsymbol{x_2}. \tag{2.1}$$

Therefore, there exists a one-to-one bijection, between the coordinates vector $\boldsymbol{x_2}$ and the parameters $\tau, \delta$ that defines the map $\boldsymbol{F}$ when $x > 0$. To show this, consider parameterising $\tau$ and $\delta$ by a single variable $v \in \mathbb{R}$ such that they are continuous functions,

$$\begin{aligned} \tau &\equiv \tau(v), \\ \delta &\equiv \delta(v). \end{aligned} \tag{2.2}$$

This parameterisation of a single variable is possible as the perturbation from some point forms a one-dimensional trajectory.

$\boldsymbol{F}^2(\mathbf{0})$ is a 2D vector function of $\tau$ and $\delta$,

$$\begin{aligned}\boldsymbol{F}^2(\mathbf{0}) &= \Big(f_1(\tau,\delta), f_2(\tau,\delta)\Big) \\ &= \Bigg(f_1\Big(\tau(v),\delta(v)\Big), f_2\Big(\tau(v),\delta(v)\Big)\Bigg).\end{aligned} \tag{2.3}$$

Note that, scalar functions $f_1$ and $f_2$ are $x$ and $y$ components of $\boldsymbol{F}^2$. They are continuous on $\mathbb{R}^2$ due to the continuity of $\boldsymbol{F}$. As $\tau(v)$ and $\delta(v)$ are also defined to be continuous, with Eq.(2.1), we see that $\boldsymbol{x_2}$ is a continuous function of $v$,

$$\boldsymbol{x_2}(v) = \boldsymbol{F}^2(\mathbf{0}). \tag{2.4}$$

To see the effects of this perturbation, one should look at the image of $P_2$, i.e. vector $\boldsymbol{x_2}$ under the map $\boldsymbol{F}$. $\boldsymbol{F}$ is dependent on the input coordinate vector $\boldsymbol{x}$ and the parameters $\tau$ and $\delta$, which are parameterised by $v$. So $\boldsymbol{F}$ can be re-expressed as

$$\boldsymbol{F} = \boldsymbol{F}(\boldsymbol{x}, v). \tag{2.5}$$

Now we should applied the initial conditions when a triangular attractor appears in Figure 1.3(a). Define $v = v_0$ such that $\boldsymbol{x_2}$ is on the $y$-axis (which means $v = v_0$ is the point when $P_2$ hits the border $x = 0$):

$$\boldsymbol{x_2}(v_0) = (0, y_2). \tag{2.6}$$

$P_2$ is mapped to $P_3$, denoted by $\boldsymbol{x_3}(v_0)$:

$$\boldsymbol{F}\left(\boldsymbol{x_2}(v_0), v_0\right) = \boldsymbol{x_3}(v_0). \tag{2.7}$$

Clearly, the left-hand side of the above equation is merely a function that is only dependent on $v_0$. Therefore, it is convenient to define a new continuous function $\boldsymbol{G}$ such that

$$\boldsymbol{G}(v) \equiv \boldsymbol{F}\left(\boldsymbol{x_2}(v), v\right). \tag{2.8}$$

So

$$\boldsymbol{G}(v_0) = \boldsymbol{x_3}(v_0). \tag{2.9}$$

Now comes the perturbation, consider $u \in \mathbb{R}$, $|u|$ is small so that $v_0 \to v_0 + u$. This subsequently changes the parameters as well as $\boldsymbol{x_2}$ and $\boldsymbol{x_3}$. Point $P_2$ moves to a new neighbouring point while for $P_3$, its coordinate vector $\boldsymbol{x_3}$, using linear approximation, gives

$$\begin{aligned}\boldsymbol{x_3}\left(v_0+u\right) &= \boldsymbol{F}\left(\boldsymbol{x_2}\left(v_0+u\right), v_0+u\right) \\ &= \boldsymbol{G}\left(v_0+u\right) \\ &= \boldsymbol{G}\left(v_0\right) + u\left.\frac{d\boldsymbol{G}}{dv}\right|_{v=v_0} \\ &= \boldsymbol{x}_3\left(v_0\right) + u\left.\frac{d\boldsymbol{G}}{dv}\right|_{v=v_0}.\end{aligned} \tag{2.10}$$

According to Eq.(2.8), $\boldsymbol{G}$ is composed of $\boldsymbol{F}$ and $\boldsymbol{x_2}$. Note that the derivative of $\boldsymbol{F}$ is discontinuous at the border as mentioned in Section 1.1.1. However, the term above, $u\frac{d\boldsymbol{G}}{dv}\big|_{v=v_0}$ is continuous with the introduction of $u$. The argument is as follows.

Take $\varepsilon > 0$ such that

$$|\boldsymbol{x_3}(v_0+u) - \boldsymbol{x}_3(v_0)| < \varepsilon. \tag{2.11}$$

From Eq.(2.10),

$$|\boldsymbol{x_3}(v_0+u) - \boldsymbol{x}_3(v_0)| = \left| u\left(\frac{d\boldsymbol{G}}{dv}\right)_{v=v_0}\right|. \tag{2.12}$$

Hence, we can always find a $\delta > 0$ such that $|u| < \delta$ and

$$\left| u\left(\frac{d\boldsymbol{G}}{dv}\right)_{v=v_0}\right| < \varepsilon \tag{2.13}$$

by letting

$$\delta < \varepsilon / \left| \left(\frac{d\boldsymbol{G}}{dv}\right)_{v=v_0}\right|. \tag{2.14}$$

Therefore, $\boldsymbol{x_3}$ is continuous at the border. This then brings a continuous change in the position of $P_3$, Thus the attractor will be continuously "deformed" in a small neighbourhood of the parameter space. This will ensure the existence of the attractor when $P_2$ is perturbed to the right or left side of the $y$-axis. We will divide our discussion into these two categories.

The above argument certainly isn't only limited to the triangular attractor but can be applied to an $n$-gon as presented in Section 1.2 by altering the number of iterations from 2 to $n$ in Eq.(2.1).

### 2.1.2 $P_2$ on the Right

First, perturb $P_2$ to the right by a small amount. The new attractor is shown in Figure 2.1. With the new set of parameters, we observed that the previous triangular attractor becomes quadrilateral.

The extra added point is $Q_2$, which is the image of $Q_1$: the intersection point of the axis with $P_2P_3$.

Start by looking at the successive images of $OP_1$, $OP_1 \to P_1P_2 \to P_2P_3$. Be aware that the image of $P_2P_3$ is NOT $P_3P_4$ because the map changes when it crosses the $y$-axis. Hence, it needs to be considered separately as two segments on each side, namely, $P_2Q_1$ and $Q_1P_3$, which maps to $P_3Q_2$ and $Q_2P_4$ respectively. Surprisingly, $P_3Q_2$ is the new boundary line for this new attractor. Note that now $P_1Q_2P_4Q_3$ must cover the fixed point to form the attractor as explained earlier.

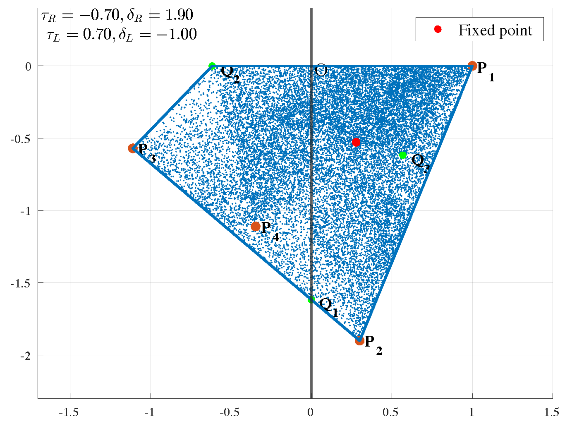


Figure 2.1: The new $P_2$ is perturbed to the right from its original position on the $y$-axis. $Q_1$ is the intersection of the line $P_2P_3$ with the axis. Under the map, $Q_1 \to Q_2 \to Q_3$, $O \to P_1 \to P_2 \to P_3 \to P_4$. $P_2$ has a new coordinate $(0.3, -1.9)$.

### 2.1.3 $P_2$ on the Left

Now perturb $P_2$ to the left of the $y$-axis. The new attractor is formed as shown in Figure 2.2. The boundary of this new attractor is not easy to find out and it's not simply the connected lines between iterated points. Before doing this, we will introduce a systematic way to find the boundary of an absorbing area, known as the *critical curve*.

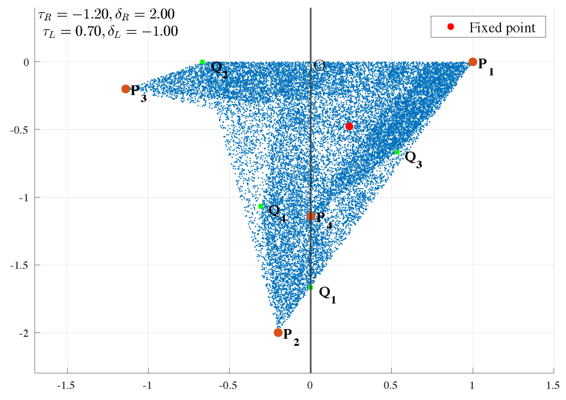


Figure 2.2: The new $P_2$ is perturbed to the left from $y$-axis. $Q_1$ is the intersection of the line $P_1P_2$ with the axis. Similar to before, $Q_1 \to Q_2 \to Q_3$, $O \to P_1 \to P_2 \to P_3 \to P_4$. $P_2$ has a new coordinate $(-0.2, -2)$.

### 2.1.4 Method of Critical Curves

Mira (1996) gave a method of studying the non-invertible maps by looking at the critical curves of the map. The border collision normal form is a non-invertible map, because the inverse of any point contains two points, each is found by two piecewise invertible maps when $x > 0$ or $x < 0$.

Given a two-dimensional non-invertible differentiable map $T : \mathbb{R}^2 \to \mathbb{R}^2$, the number of preimages of a point $(x, y)$ maybe 0 or 1, 2,.... Hence it is natural to divide the plane $\mathbb{R}^2$ into regions $Z_0, Z_1, ..., Z_n$, where $Z_k$, for some $k$, denotes the set of points having exactly $k$ preimages. As point crosses the boundaries of different $Z_k$, preimages are generated or destroyed. This gives rise to the definition of critical curve $LC$ as the set of points having two or more coincident preimages.

Given an absorbing area $A$, define the segment $\gamma$ as

$$\gamma = A \cap LC. \tag{2.15}$$

The boundary $\partial A$ of the absorbing area $A$, when $A$ is invariant, is found by

$$\partial A \subset \bigcup_{k=1}^{m} T^k(\gamma) \tag{2.16}$$

for some suitable integer $m$.

In the case when $A$ is not invariant, the invariant absorbing area $\mathcal{I}$ is found by finding the intersections of finite iterate of $A$:

$$\mathcal{I} = \bigcap_{k=1}^{m} T^k(A) \tag{2.17}$$

for some finite $m$.

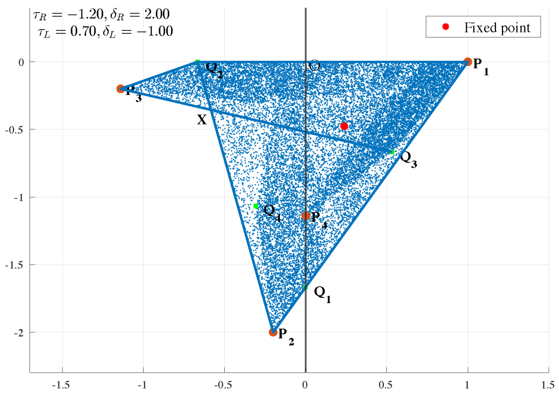


Figure 2.3: Boundary of the attractor is formed as $P_1P_2XP_3Q_2$. $X$ is the intersection of $P_3Q_3$ and $Q_1P_2$.

Now looking back at Figure 2.2. The border, $y$-axis, can be regarded as a critical curve. Since the number of preimages of some point on the axis is exactly 1, whereas the rest points give 2 preimages. We look at its intersection of the $y$-axis with the absorbing area to get the line $\gamma = OQ_1$ and track down its images under the map for some finite number of iterates to get a union of lines following the process in Eq.(2.16). The boundary is then a subset of the newly formed union of lines and we will find the exact boundary by inspecting the computer-generated graph.

Inspect the images of $OQ_1$:

$$OQ_1 \to P_1Q_2 - \left[ \begin{array}{l} OP_1 \to P_1P_2 - \left[ \begin{array}{l} P_1Q_1 \to P_2Q_2 \to P_3Q_3 \\ P_2Q_1 \to P_3Q_2 \to P_4Q_3 \end{array} \right. \\ OQ_2 \to P_1Q_3 \to P_2Q_4 \end{array} \right. \tag{2.18}$$

The above procedure stops once we find all the bounding curves. Notice that the image of some lines has two parts because it crosses the border and needs to be mapped separately.

Combine the above iterated lines with Figure 2.2, and take the union of these lines, the boundary is found to be a polygon with 5 sides $P_1P_2XP_3Q_2$ as shown in Figure 2.3.

## 2.2 Bifurcation On the Boundary

We should now proceed from bifurcation on the border and treat the generated attractors as the starting point of new bifurcations. In the previous sections, i.e. in Figures 2.3 and 2.1, the perturbed point $P_2$ affects only the parameters on the right $R$ when $x > 0$. The parameters on the $L$, $\tau_L$ and $\delta_L$ are set to be fixed and the fixed point on $R$ is in the polygon. Therefore, we should let $Q_3$, the image of $Q_2$, cross the boundary $P_1P_2$, and check what would happen to the geometry of the attractor.

### 2.2.1 When $P_2$ is On the Right

First, consider the case when $P_2$ is on the right (See Figure 2.1). Let $Q_3$, the image of $Q_2$ (which is the image of $Q_1$, where $P_2P_3$ intersects the axis), cross the boundary line $P_1P_2$ by a small amount. Figure 2.4 was generated.

Follow the same procedure in Eq.(2.18):

$$OQ_1 \to P_1Q_2 - \left[ \begin{array}{l} OP_1 \to P_1P_2 \to P_2P_3 \\ OQ_2 \to P_1Q_3 \to P_2Q_4 - \left[ \begin{array}{l} P_2R_1 \to P_3R_2 \\ Q_4R_1 \end{array} \right. \end{array} \right.$$

From this we get the new polygonal attractor with 8 sides: $R_2P_1R_3X_1P_2R_4X_2P_3$. 4 extra sides are added. Intuitively, this is because when $Q_3$ escapes $P_1P_2$, an extra area $P_1R_3X_1$ leaks out of the initial quadrilateral which adds two sides. This extra area is then mapped to $P_2R_4X_2$ which adds another two sides.

Now we examine other potential bifurcations. $P_1, P_2, P_3$ form the vertices of the polygon. Once $P_4$, the image of $P_3$, goes inside the polygon, it never escapes because it is an absorbing area. Figure 2.5 shows what would happen if we break this criterion, i.e. $P_4$ crosses the line $P_1P_2$.

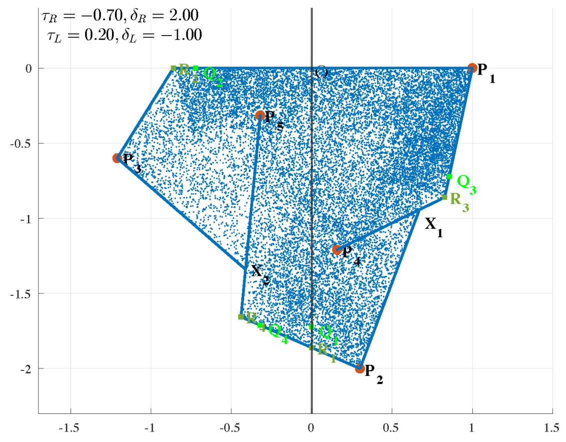


Figure 2.4: $Q_3$ crosses the initial boundary line $P_1P_2$ by a small amount. $X_1$ is the intersection of $P_1P_2$ and $R_3P_4$. $X_2$ is the intersection of $P_2P_3$ and $R_4P_5$. $X_1$ is mapped to $X_2$.

Again, use the method of critical curves to find the boundary. Track down the images of $OQ_1$:

$$OQ_1 \to P_1Q_2 - \left[\begin{array}{l} OP_1 \to P_1P_2 \to P_2P_3 \\ OQ_2 \to P_1Q_3 \to P_2Q_4 \end{array}\right.$$

$$\begin{array}{l} P_2P_3 - \left[\begin{array}{l} P_2Q_1 \to P_3Q_2 \to P_4Q_3 \\ P_3Q_1 \to P_4Q_2 \end{array}\right. \\ P_2Q_4 - \left[\begin{array}{l} P_2R_1 \to P_3R_2 \to P_4R_3 \to P_5R_4 \\ Q_4R_1 \to Q_5R_2 \to Q_6R_3 \end{array}\right. \end{array}$$

This new attractor: $P_1R_3P_4X_1P_2R_4P_5R_5X_2P_3R_2$ has 11 sides, 3 more compared to Figure 2.4.

A question arises when both $Q_3$ and $P_4$ cross the line $P_1P_2$. Which point crosses the $P_1P_2$ first? Is it always $Q_3$? If not, then is there a new bifurcation when $P_4$ comes out and $Q_3$ is inside? The answer is NO. $P_4$ cannot exit the boundary if $Q_3$ doesn't exit. A simple proof is given as follows using explicitly the border collision normal form.

## Proving the order of exiting $P_1P_2$

Recall the normal form in Eq.(1.2) and look at its iterated points $O \to P_1 \to P_2 \to P_3 \to P_4$, $Q_1 \to Q_2 \to Q_3$, $A \to B$, as illustrated in Figure 2.6. Suppose $Q_3$ is on the line $P_1P_2$. Then, to see if $P_4$ exits the line, one simply needs to compare the slope of the lines $P_1P_2$ and $P_1P_4$. The equation of $P_1P_2$ is

$$y = -\frac{\delta_R}{\tau_R}(x-1). \tag{2.19}$$

So its slope is simply $-\delta_R/\tau_R$. Now we should determine the condition that $Q_3$ is on the line $P_1P_2$. But the algebraic expression of $Q_3$ is quite complicated.

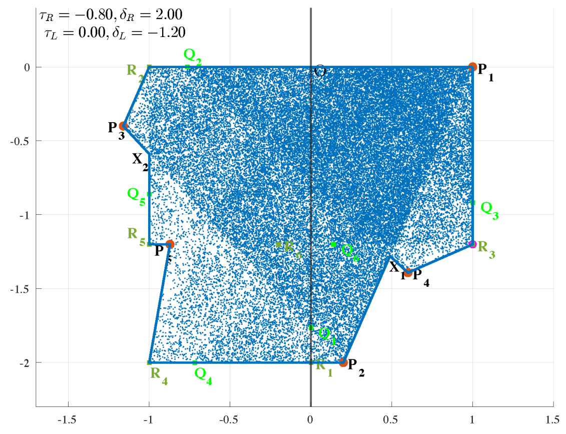


Figure 2.5: Generated new attractor when both $Q_3$ and $P_4$ cross the line $P_1P_2$. $X_1$ is the intersection of $P_1P_2$ and $Q_2P_4$. $X_2$ is the intersection of $P_2P_3$ and $R_2R_5$.

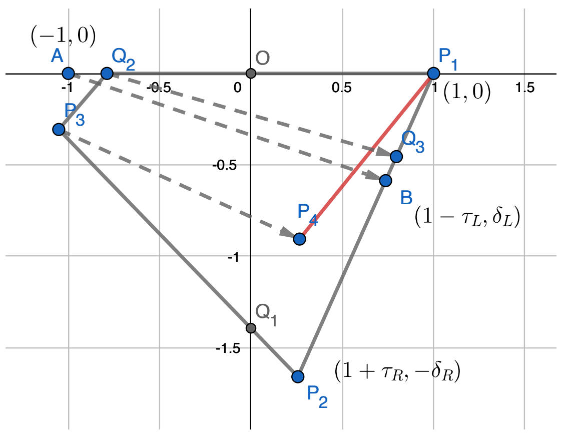


Figure 2.6: A schematic diagram showing the setup for the proof.

To avoid this, we can choose a point $A(-1, 0)$ which is near $Q_2$ because the bifurcations always happen in a small neighbourhood of parameter space. $B(1 - \tau_L, \delta_L)$ is the image of $A$. If $Q_3$ is on $P_1P_2$, $B$ must also be on $P_1P_2$ ($Q_3$ and $B$ are close and both of them are on negative $x$-axis). Plug the coordinate of $B$ in Eq.(2.19), we get the condition that $B$($Q_3$, equivalently) is on $P_1P_2$:

$$\frac{\delta_L}{\tau_L} = \frac{\delta_R}{\tau_R}. \tag{2.20}$$

The coordinate of $P_4$ is

$$\begin{pmatrix} -\delta_R(\tau_L + \tau_R + 1) + \tau_L(\tau_R^2 + \tau_R + 1) + 1 \\ -\delta_L(-\delta_R + \tau_R^2 + \tau_R + 1) \end{pmatrix}. \tag{2.21}$$

Substitute the above condition to the coordinate of $P_4$, we get a new coordinate and the slope of $P_1P_4$ can be computed, call it $k_1$. Denote the slope of $P_1P_2$

as $k_0 \equiv -\delta_R/\tau_R$. If we can show that the slope of $P_1P_4$ is always smaller than $P_1P_2$, i.e., $k_1/k_0 < 1$, then it is proved. And if

$$\frac{k_1}{k_0} = \frac{\tau_L\left(\delta_R - \tau_R\left(\tau_R + 1\right) - 1\right)}{\delta_R\left(\tau_L + \tau_R + 1\right) - \tau_L\left(\tau_R^2 + \tau_R + 1\right)} < 1, \tag{2.22}$$

this expression can then be simplified to

$$\delta_R(1 + \delta_R) > 0. \tag{2.23}$$

This always holds because $\delta_R > 0$ (we assumed $P_2$ always has a negative $y$-component).

□

### 2.2.2 When $P_2$ is On the Left

Start from the attractor in Figure 2.3 when Let $Q_3$ exits the line $P_1P_2$ by a small amount, but $P_4$ is still inside. As explained in the last section, different bifurcations will occur when $P_4$ is also outside and if $P_4$ is outside, $Q_3$ must be outside as well. The plot in Figure 2.7 is observed.

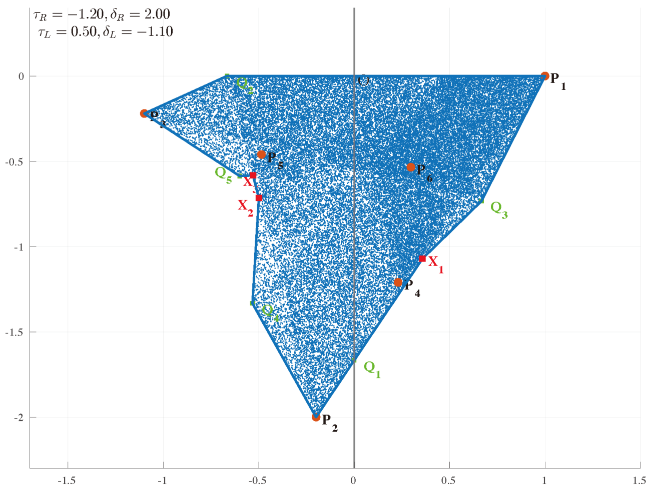


Figure 2.7: $Q_3$ exits the line $P_1P_2$. A new attractor with 10 sides is formed. $X_1$ is the intersection of $P_1Q_1$ and $Q_3P_4$. $X_2$ is the intersection of $P_2Q_2$ and $Q_4P_5$. $X_3$ is the intersection of $P_2Q_2$ and $Q_5P_6$. $X_1$ maps to $X_2$.

Boundary lines are found as usual by:

$$OQ_1 \to P_1Q_2 - \left[\begin{array}{l} OP_1 \to P_1P_2 \\ OQ_2 \to P_1Q_3 \to P_2Q_4 \to P_3Q_5 \end{array}\right.$$

$$P_1P_2 - \left[\begin{array}{l} P_1Q_1 \to P_2Q_2 \to P_3Q_3 \\ P_2Q_1 \to P_3Q_2 \to P_4Q_3 \to P_5Q_4 \to P_6Q_5 \end{array}\right.$$

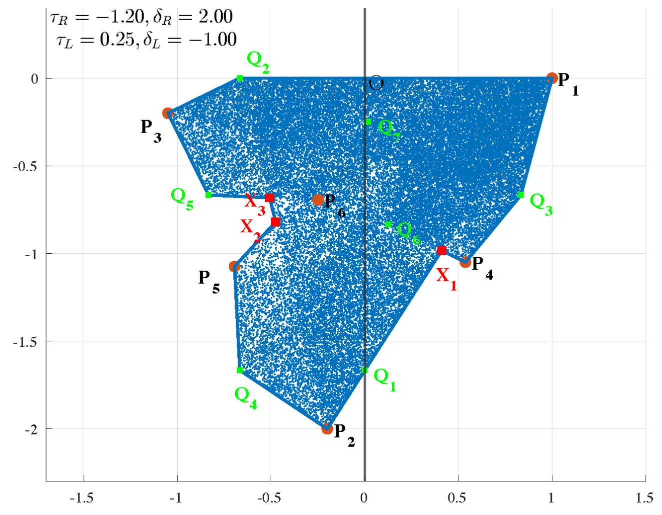


Figure 2.8: Both $P_4$ and $Q_3$ exit the boundary line $P_1P_2$. $X_1$ is the intersection of $P_1Q_1$ and $Q_6P_4$. $X_2$ is the intersection of $P_2Q_2$ and $Q_7P_5$. $X_3$ is the intersection of $P_2Q_2$ and $Q_5P_6$. $X_1$ maps to $X_2$. This attractor is a 12-gon.

Now let $P_4$ cross the line as well. We get Figure 2.8.

To find the extra two sides added compared to Figure 2.7, we need to iterate a few more times on the line $P_3Q_5$:

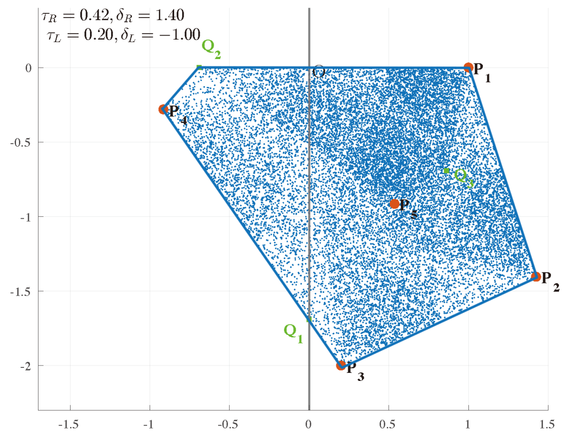
$$P_3Q_5 \to P_4Q_6 \to P_5Q_7$$

## 2.3 More Bifurcations

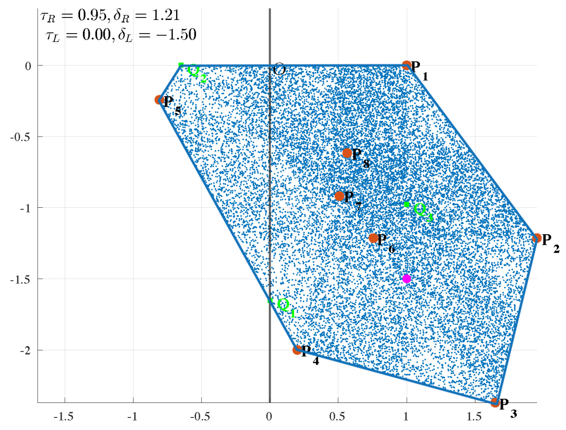
In the previous discussion, we have only observed the cases of bifurcations initiated from a triangular attractor. However, as we have explained in Section 2.1.1 using the continuity argument, this is not restricted to the number of sides. In Section 1.2, it is shown that polygon attractors can be constructed with increasing number of sides.

Here, we illustrate a few more example figures when we get more number of sides. For instance, Figure 2.9 shows the existence of attractors with more sides when the point moves away from the $y$-axis by a small amount. Figure 2.10 shows the bifurcation on the border for a quadrilateral attractor and Figure 2.11 shows the bifurcation on the boundary. These are compatible with the expectation.

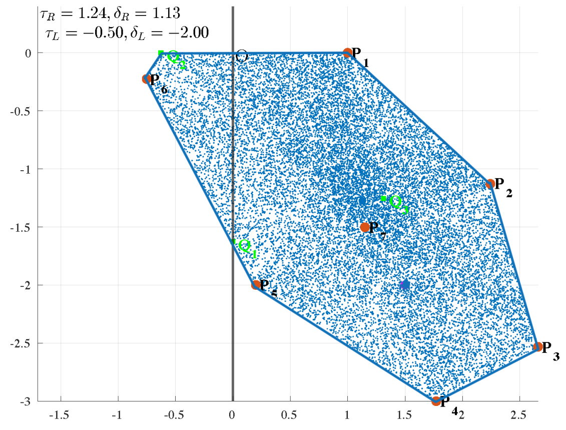
In all of these setups, one should always note the assumptions we made for constructing the attractor in the previous chapters. And all bifurcations occur in a small neighbourhood of the parameter space from the initial state. If it is not small, many constraints would be violated and complicated geometry may appear.



(a)



(b)



(c)

Figure 2.9: Attractors formed by perturbing the point a small amount from the $y$-axis with increasing number of sides.

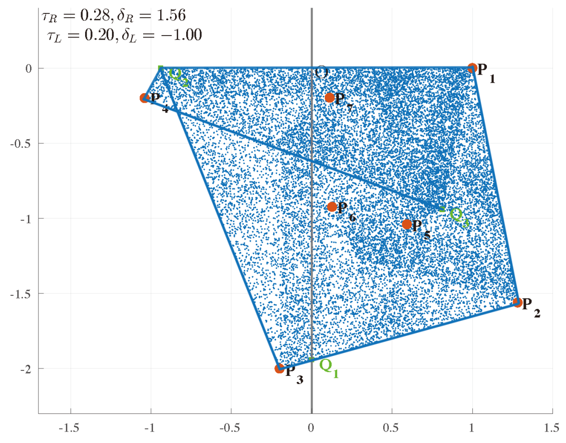


Figure 2.10: $P_3$ is perturbed to the left.

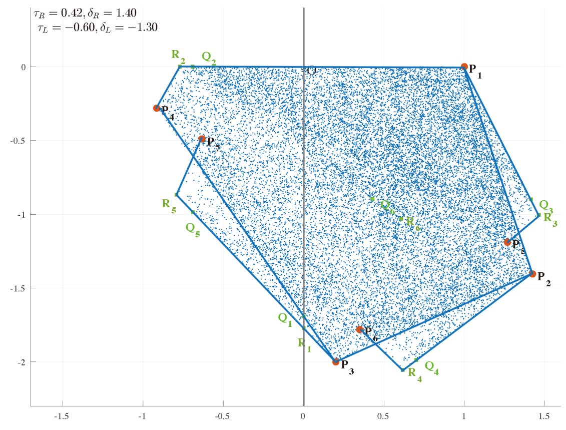


Figure 2.11: $P_3$ is perturbed to the right while $Q_3$ also exits $P_1P_2$.

# Chapter 3

# Conclusion

We introduced the concepts of absorbing area and attractor in piecewise smooth systems, which can be transformed to a border collision normal form up to affine linear approximation. Then we come up with the assumptions used to construct the simple polygonal attractors and created a compatible code to numerically assist the analysis in the bifurcations of the geometry of the attractors. The bifurcation occurs when we have a small perturbation in its parameter space that either causes the point to cross the border or certain boundary lines. By a continuity argument, the existence of such an attractor in a small neighbourhood of the parameter space is ensured. The possible bifurcations in the geometry of the attractors were analysed with the help of Matlab programs. A mechanism is derived from the method of critical curves to find the boundary of the attractors.

Due to the very limited amount of time provided for this project, many proposed ideas were not studied:

- The assumptions made for generating such a simple attractor could only be a subset of a more general space of parameters. If given more time, we wish to find out such parameter space.
- A rigorous proof for the generation of the discussed attractors is not given. In (Glendinning and Wong; 2011), this is done by first proving it is invariant, then creating a basin that contains the invariant absorbing area to prove it is attracting. If given more time, we would like to explore the possibility of a general proof.
- The study relied heavily on the use of computer. If we can devise a better automatic program to dynamically visualise the behaviour when changing the parameters, we may have more insightful results.
- The border collision normal form is a continuous map in $\mathbb{R}^2$, thus, is it possible to generalise and induce a bifurcation theory for any continuous two-dimension map.
- Classify all possible bifurcations in the geometry of the attractors.

In the end, we get a series of bifurcations of the attractors. One could classify them into the product of these two cases:

1. The point $P_n$ is either on the right or left.
2. Whether the point on the negative $x$-axis crosses the boundary line.

If we increase the number of sides of the initial attractors before the perturbation, the bifurcations become more complicated. More vertices and more sides appeared, It may occur that beyond a certain point when the number of sides of the initial attractor $n = n_0$, there no longer exists a "simple" polygonal attractors. Because

1. More sides are iterated and vertices come extremely close to each other that it is difficult to distinguish between vertices and we may not have simple lines that form the boundary of the attractor.
2. More iterations will make many conditions become significantly sensitive to the initial condition (or perturbation). Therefore, the intersection of these neighbourhoods of the parameter space may be an empty set and there no longer exists an attractor at all.

The possible conclusion from this is that for a two dimensional piecewise smooth, everywhere continuous map, we may only find a polygonal attractor for a finite number of sides.

# Bibliography

Banerjee, S., Yorke, J. A. and Grebogi, C. (1998). Robust chaos, *Physical Review Letters* **80**(14): 3049.

Di Bernardo, M., Budd, C. J., Champneys, A. R., Kowalczyk, P., Nordmark, A. B., Tost, G. O. and Piiroinen, P. T. (2008). Bifurcations in nonsmooth dynamical systems, *SIAM review* **50**(4): 629–701.

Glendinning, P. and Wong, C. H. (2011). Two-dimensional attractors in the border-collision normal form, *Nonlinearity* **24**(4): 995.

Mira, C. (1996). *Chaotic dynamics in two-dimensional noninvertible maps*, Vol. 20, World Scientific.

Nusse, H. E. and Yorke, J. A. (1992). Border-collision bifurcations including "period two to period three" for piecewise smooth systems, *Physica D: Nonlinear Phenomena* **57**(1-2): 39–57.

# Appendix

## main.m

```
%% INITIAL PARAMETERS
% set_p0             position of the orgin
% nr                 # of iterations on the RIGHT (x>0)
% nit                # of iterations for many points
% set_y              assigned y-coord on x-axis
% set_pt_nr          assigned position after *nr* iterations on the
    RIGHT (x>0)
% set_pt_leftmapped assigned position by the map on the LEFT (x<0)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;
clear global;
global tr dr tl dl set_y
figure;
nr=5;
nit=20000;
set_p0=[0;0];
set_y=-2;
set_pt_nr=[-.2;set_y];
set_pt_leftmapped=[.5;-1];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%% SYMBOLIC CALCULATION USED FOR vpasolve
var_r=set_p0;
for i = 1:nr
    var_r = sym_fr(var_r);%%%%%
end

var_l = sym_fl([set_y+1;0]);%%%


%% FIND *tr *dr* *tl* *dl*
% the solution from _vpasolve_ may contain complex, exclude them
% (what does complex represent???)
% @@@ *tr* and *dr* may be multi-dimension arrays (multiple
    solutions)

[dr,tr]=vpasolve(var_r == set_pt_nr);
[dr,tr]=deal(double(dr(imag(dr)==0)),double(tr(imag(tr)==0)));
[dr,tr]=deal(dr(1),tr(1));
% [dr,tr]=deal(dr+0 ,tr-0.1);

```

```
[dl,tl]=vpasolve(var_l == set_pt_leftmapped);
[dl,tl]=deal(double(dl),double(tl));

%% Filter tr dr
% all iterated points before *p_nr* needs to be on the right
% use this to exclude some tr dr that may carry points to the left

%% PLOT ITERATIONS AND FIRST FEW LINES
iter_pt=set_p0;
ptP=[];
for i=1:nit %collect points
        iter_pt=f(iter_pt);
        ptP=[ptP,iter_pt];
end

scatter(ptP(1,:),ptP(2,:),...
    10,[0 0.4470 0.7410],'.'); % iteration plot
hold on

% polygon=[ptP(:,1:nr+1),ptP(:,1)];
% plot(polygon(1,:),polygon(2,:),'Color',[0 0.4470 0.7410],'LineWidth',2);
% hold on
plot(ptP(1,1:nr+2),ptP(2,1:nr+2),".",'Marker','o',...
    'MarkerFaceColor',[0.8500 0.3250 0.0980],...
    'MarkerEdgeColor',[0.8500 0.3250 0.0980],'Color','k','LineWidth',2); % first few lines


for i=1:nr+2
    text(ptP(1,i)+0.03,ptP(2,i)-0.03,...
        sprintf('P_{%d}',i),...
        'Color','k',...
        'FontSize',14,'FontWeight','bold');
end



%% PLOT FIXED POINTS AND LEFT MAPPED POINT
fixed_pt_r=1/(1+dr-tr)*[1;-dr];
plot_left_mapped=scatter(set_pt_leftmapped(1),set_pt_leftmapped(2),...
    50,'m','filled');
% hold on
% plot_fixed=scatter(fixed_pt_r(1),fixed_pt_r(2),...
%     50,'r','filled');


%% ANNOTATIONS & TEXTS & PLOT RANGE ...
grid on
xlim([-1.7 1.5]);
ylim([-2.3 0.4]);
xLim=xlim;
yLim=ylim;
str_in_plot=sprintf("$\\tau_R=%1.2f,\\delta_R=%2.2f$\n $\\tau_L=%3.2f,\\delta_L=%4.2f$",...
    tr,dr,tl,dl);
text(0.99*xLim(1),0.99*yLim(2),str_in_plot,...
    'HorizontalAlignment','left','VerticalAlignment','top',...
    'Interpreter','latex','FontSize',14);
text(0.02,-0.02,'O','FontSize',16)
% title(sprintf('title'),'FontSize',16,'FontWeight','bold')
```

```
xline(0,'LineWidth',1.5);
% legend( plot_fixed,...
%      'Fixed point','FontSize',14);



%% DEFINE FUNCTION
function x2 = sym_fr(x)
syms tr dr
m=[1;0];
A_R=[tr 1;-dr 0];
x2=A_R*x+m;
end

function x2 = sym_fl(x)
syms tl dl
m=[1;0];
A_L=[tl 1;-dl 0];
x2=A_L*x+m;
end
```

## label.m

```
% figure;
[n1,n2]=deal(3,3);

pt_start=InterX(ptP(:,5:6),...
    [0,0;0,-5]);
ptQ=[];
for i=1:n1 %collect points
    ptQ=[ptQ,pt_start];
    pt_start=f(pt_start);
end

pt_start=InterX([ptP(:,3),ptQ(:,5)],[0,0;0,-5]);
ptR=[];
for i=1:n2 %collect points
    ptR=[ptR,pt_start];
    pt_start=f(pt_start);
end

X1=InterX([ptQ(:,6),ptP(:,4)],[ptQ(:,1),ptP(:,1)]);
X2=InterX([ptQ(:,7),ptP(:,5)],[ptQ(:,2),ptP(:,2)]);
X3=InterX([ptQ(:,5),ptP(:,6)],[ptQ(:,2),ptP(:,2)]);

%% PLOT, LABEL
polygon=[ptQ(:,2),ptP(:,1),ptQ(:,3),ptP(:,4),X1,...
    ptP(:,2),ptQ(:,4),ptP(:,5),X2,X3,...
    ptQ(:,5),ptP(:,3),ptQ(:,2)];
plot(polygon(1,:),polygon(2,:),'Color',[0 0.4470 0.7410],'LineWidth
    ',2);

scatter(ptQ(1,:),ptQ(2,:),...
    36,'g','s','filled');
scatter(ptR(1,:),ptR(2,:),...
    36,[0.4660 0.6740 0.1880],'s','filled');
scatter([X1(1),X2(1),X3(1)],[X1(2),X2(2),X3(2)],...
    50,'r','s','filled');


for i=1:n1
```

```
    text(ptQ(1,i)+0.05,ptQ(2,i)-0.05,...
        sprintf('Q_{%d}',i),...
        'Color','g',...
        'FontSize',14,'FontWeight','bold');
end
for i=1:n2
    text(ptR(1,i)+0.03,ptR(2,i)-0.03,...
        sprintf('R_{%d}',i),...
        'Color',[0.4660 0.6740 0.1880],...
        'FontSize',14,'FontWeight','bold');
end

text(X1(1)+0.03,X1(2)-0.03,'X_1',...
        'Color','r',...
        'FontSize',14,'FontWeight','bold');
text(X2(1)+0.03,X2(2)-0.03,'X_2',...
        'Color','r',...
        'FontSize',14,'FontWeight','bold');
text(X3(1)+0.03,X3(2)-0.03,'X_3',...
        'Color','r',...
        'FontSize',14,'FontWeight','bold');
```

## f.m

```
function output=f(x)
    % return either the image of a line or a point
    % input a 2 by 1 column vector: [x1;y1]
    % return the image: 2 by 1 column vector, under the map
    % input a 2 by 2 column vector: [x1,x2;y1,y2]
    % see if x1*x2>=0
    % if so, means two points are on one side, no intersection with
     x=0
    % return a 2 by 2 column vector consisting the image point of
    each
    % column vector
    % if not, then x1*x2<0, there's intersection point
    % return a 2 by 3 column vector corresponding to the image of
    each pt
    global set_y
    n_col=size(x,2);
    switch n_col
        case 1
            output=f_pt(x);
        case 2
            if x(1,1) * x(1,2)>=0
                output=[f_pt(x(:,1)),f_pt(x(:,2))];
            else
                itsc=InterX([x(1,:);x(2,:)],[0,0;0,set_y]);
                output=[f_pt(x(:,1)),f_pt(itsc),f_pt(x(:,2))];
            end
    end

    function im_pt = f_pt(x)
        global tl dl tr dr
        m=[1;0];
        A_L=[tl 1;-dl 0];
        A_R=[tr 1;-dr 0];
        if x(1)>=0
            im_pt=A_R*x+m;
        else
            im_pt=A_L*x+m;
        end
    end
end
```

## InterX.m

```
function P = InterX(L1,varargin)
%INTERX Intersection of curves
%   P = INTERX(L1,L2) returns the intersection points of two curves
    L1
%   and L2. The curves L1,L2 can be either closed or open and are
    described
%   by two-row-matrices, where each row contains its x- and y-
    coordinates.
%   The intersection of groups of curves (e.g. contour lines,
    multiply
%   connected regions etc) can also be computed by separating them
    with a
%   column of NaNs as for example
%
%         L  = [x11 x12 x13 ... NaN x21 x22 x23 ...;
%               y11 y12 y13 ... NaN y21 y22 y23 ...]
%
%   P has the same structure as L1 and L2, and its rows correspond
    to the
%   x- and y- coordinates of the intersection points of L1 and L2.
    If no
%   intersections are found, the returned P is empty.
%
%   P = INTERX(L1) returns the self-intersection points of L1. To
    keep
%   the code simple, the points at which the curve is tangent to
    itself are
%   not included. P = INTERX(L1,L1) returns all the points of the
    curve
%   together with any self-intersection points.
%
%   Example:
%       t = linspace(0,2*pi);
%       r1 = sin(4*t)+2;  x1 = r1.*cos(t); y1 = r1.*sin(t);
%       r2 = sin(8*t)+2;  x2 = r2.*cos(t); y2 = r2.*sin(t);
%       P = InterX([x1;y1],[x2;y2]);
%       plot(x1,y1,x2,y2,P(1,:),P(2,:),'ro')

%   Author : NS
%   Version: 3.0, 21 Sept. 2010

%   Two words about the algorithm: Most of the code is self-
    explanatory.
%   The only trick lies in the calculation of C1 and C2. To be
    brief, this
%   is essentially the two-dimensional analog of the condition that
    needs
%   to be satisfied by a function F(x) that has a zero in the
    interval
%   [a,b], namely
%           F(a)*F(b) <= 0
%   C1 and C2 exactly do this for each segment of curves 1 and 2
%   respectively. If this condition is satisfied simultaneously for
    two
%   segments then we know that they will cross at some point.
%   Each factor of the 'C' arrays is essentially a matrix containing
%   the numerators of the signed distances between points of one curve
%   and line segments of the other.

    %...Argument checks and assignment of L2
    error(nargchk(1,2,nargin));
    if nargin == 1,
        L2 = L1;     hF = @lt;    %...Avoid the inclusion of common points
    else
        L2 = varargin{1}; hF = @le;
    end

    %...Preliminary stuff
    x1  = L1(1,:)';  x2 = L2(1,:);
    y1  = L1(2,:)';  y2 = L2(2,:);
    dx1 = diff(x1); dy1 = diff(y1);
    dx2 = diff(x2); dy2 = diff(y2);

    %...Determine 'signed distances'
    S1 = dx1.*y1(1:end-1) - dy1.*x1(1:end-1);
    S2 = dx2.*y2(1:end-1) - dy2.*x2(1:end-1);

    C1 = feval(hF,D(bsxfun(@times,dx1,y2)-bsxfun(@times,dy1,x2),S1),0);
    C2 = feval(hF,D((bsxfun(@times,y1,dx2)-bsxfun(@times,x1,dy2))',S2'),0)';

    %...Obtain the segments where an intersection is expected
    [i,j] = find(C1 & C2);
    if isempty(i),P = zeros(2,0);return; end;

    %...Transpose and prepare for output
    i=i'; dx2=dx2'; dy2=dy2'; S2 = S2';
    L = dy2(j).*dx1(i) - dy1(i).*dx2(j);
    i = i(L~=0); j=j(L~=0); L=L(L~=0);  %...Avoid divisions by 0

    %...Solve system of eqs to get the common points
    P = unique([dx2(j).*S1(i) - dx1(i).*S2(j), ...
                dy2(j).*S1(i) - dy1(i).*S2(j)]./[L L],'rows')';

    function u = D(x,y)
        u = bsxfun(@minus,x(:,1:end-1),y).*bsxfun(@minus,x(:,2:end),y);
    end
end
```