# Stability Analysis on Replicator Games with Perturbations

B10901042
彭瑆 (Wen Perng)
*Department of Electrical Engineering, National Taiwan University*

December 25, 2022

---

## *Abstract*

In this study, we delve into the details of replicator dynamics. Replicator dynamic is a repeated game in which one player "copies" other's strategies if he finds them with higher payoffs.

We provide a detailed derivation of the system dynamics of the game, creating a phase portrait of it, and uses it to analyze the stability of the Nash equilibria present. Three types of Nash equilibria exist: diverging, oscillatory stable, and asymptotically stable. Thus, with the phase portrait, we can obtain more information about the game in comparison with only algebraic manipulations on the game matrix. Moreover, we can see how it evolves over time.

And last but not least, we analyze the effect of payoff perturbations on the stability of a game. Furthermore, a physical interpretation of the perturbation was given by considering how a player reacts to unexpectedness.

## *Acknowledgements*

Special thanks to professor Shi-Chung Chang for his questions during the course–Information, Control and Games (111-1) that inspired this study.

And much appreciation to a friend of mine, Yao-Yu Lee. Through discussions with him, I was able to overcome hurdles throughout the project, and came up with the solutions to the questions regarding perturbations in payoff.

# 1 Motivation and Objectives of Study

## 1.1 Motivation and Inspirations

Replicator dynamics is a strategy to employ while playing a repeated game within a population of players. Simply by copying others' action, one expects to achieve a higher payoff. Such creates a nonlinear dynamic, and thus induce many nontrivial results.

Why do these dynamics occur? How do we quantize them? These are the questions we would like to touch on in this study.

Moreover, replicator dynamics provides insight into the different strategic equilibria (pure Nash and mixed). It allows us to know more about how the game evolves over time, and how to manipulate it.

Many great examples are suitable for us to study. One of which is the Shapley's rock-paper-scissors game, in which players' strategies oscillate between the three choices and never reaches the mixed strategy equilibrium. Another is the game of chicken, or the battle of sexes, many of which are suitable for modeling current events.

## 1.2 Objectives

In linear systems, we study systems of equations with the so-called state-space model. Though the game dynamics we are studying is not linear, we can focus on the neighborhood of a Nash equilibrium (a stationary point) of the system, and linearize in this neighborhood. Linearization provides us with the powerful tools of stability analysis and simple phase portrait analysis.

Moreover, in [2], we saw that by adding a perturbation (in terms of the paper, it uses Gibbs entropy) to the payoff, an oscillating game can converge to the unique mixed Nash equilibrium. Many questions arose, too: Which perturbations are suitable for the strategies to converge? Why do the strategies converge? And most importantly, what are the physical importance/meaning of such perturbations?

# 2 Replicator Dynamics

Replicator dynamics is a game in which a population (or more) of decision players randomly (uniformly) compete with each other. If ones payoff is lower than his opponent, he will change to other's strategy proportionally. Such simple learning strategy creates rich interpretations and many non-trivial results.

The simple setting of a replicator game is as follow: suppose players in a population all have the same strategy set $S$, with $n$ different actions/pure strategies. The population as a whole have $x_k \in [0,1]$ $(k = 1, 2, \cdots, n)$ of the total population playing action $s_k$ at a given time $t$. We denote the the *state* of the whole populace as:

$$\boldsymbol{x} = [x_1 \; x_2 \; \cdots \; x_n]^\mathsf{T},$$

also called the *strategy* of the population. It is a *standard simplex*, denoted also as

$$\boldsymbol{x} \in \Delta(n) = \Big\{ [x_1 \; x_2 \; \cdots \; x_n]^\mathsf{T} \, | x_k \geq 0, \; x_1 + x_2 + \cdots + x_n = 1 \Big\}. \tag{2.1}$$

When will one player copy another player's action? Due to the uncertainty of information and the stochastic nature of the game at play, one switches action if the expected profit is larger. I.e.,

$$\dot{x}_k(t) \propto - \Big[ \boldsymbol{x}^\mathsf{T} \Pi \boldsymbol{x} - (\Pi \boldsymbol{x})_k \Big].$$

Where $\Pi \in \mathbf{R}^{n \times n}$ is the game matrix, and the element $[\Pi]_{ij}$ denotes the payoff for a player if his action is $s_i$ and his opponent has action $s_j$. $(\Pi \boldsymbol{x})_k$ (the $k^{\text{th}}$ element of the vector) is the expected payoff for a player with action $s_k$, and $\boldsymbol{x}^\mathsf{T} \Pi \boldsymbol{x}$ is the average payoff of the whole populace. Moreover, the change in population playing $s_k$ is directly proportional to the number of players playing $s_k$. Thus, we have

$$\dot{x}_k = \alpha x_k \Big[ (\Pi \boldsymbol{x})_k - \boldsymbol{x}^\mathsf{T} \Pi \boldsymbol{x} \Big] \tag{2.2}$$

with $\alpha > 0$ being a constant, and $\alpha < 0$ is equivalent to $\Pi$ being the cost instead of payoff. The set of nonlinear differential equations (2.2) is known as the *replicator dynamics*.

A more rigorous derivation from the point of view of each player is presented in the next subsection, along with a detailed introduction to replicator dynamics.

## 2.1 Definitions and Game Settings of Replicator Dynamics

### 2.1.1 Derivation of Replicator Dynamics

Replicator dynamics model a *population with frequent interactions*, in which every player plays a pure strategy, and at every stage of the game, we pair off all agents to obtain some payoff. The dynamics tells us how portion of various pure strategies evolve over time.

Some settings are as below:

- $\sigma$: the whole population.
- If the pair-off are in groups of two, the players play a 2-player normal form symmetric game:
    1. Strategies: $S = \{s_1, s_2, \cdots, s_n\}$ are the strategies of each player.
    2. Payoff: $\pi_1(s_i, s_j)$, payoff of player 1 playing $s_i$ while opponent plays $s_j$.
    3. Symmetric Payoff: $\pi_1(s_i, s_j) = \pi_2(s_j, s_1) \coloneqq \pi_{ij}$, represented in matrix $\Pi$.
    4. Symmetric on Strategy: can not condition strategy on whether one is player 1 or 2.
    5. Information: at each round, the information of each player are his payoff, the payoff of his opponent, and the action of his opponent.
- Game repeated at every $\mathrm{d}t$ interval.
- $x_i(t)$: fraction of players in $\sigma$ playing $s_i$ at stage $t$. The expected payoff of one player to playing $s_i$ is

$$\pi_{i\sigma}(t) = \sum_{j=1}^{n} \pi_{ij} x_j(t) = (\Pi \boldsymbol{x})_i \,.$$

- Index the strategies such that $\pi_{i\sigma} \leq \pi_{(i+1)\sigma}(t)$.

Next, the dynamics are described as follow:

- In $\mathrm{d}t$, player learns a random agent's (his opponent) payoff with probability $\alpha_1 \mathrm{d}t > 0$.
- Replicate opponent's strategy if opponent's payoff is higher.

- However, information concerning the difference in payoff is imperfect, thus, a larger difference in payoff, the more likely one is to perceive it.
- The expected probability that one playing $s_i$ switches to $s_j$ is:

$$x_{i,j}(t) = \begin{cases} \alpha_2 \left[\pi_{j\sigma}(t) - \pi_{i\sigma}(t)\right] & , \pi_{i\sigma}(t) < \pi_{j\sigma}(t) \\ 0 & , \pi_{i\sigma}(t) \geq \pi_{j\sigma}(t) \end{cases} \tag{2.3}$$

  And $\alpha_2$ is small enough such that $x_{i,j} \leq 1$ always holds.

The fraction of players choosing $s_i$ in $t + \mathrm{d}t$ will thus be:

$$\begin{aligned} x_i(t+\mathrm{d}t) =& x_i(t) + \alpha_1 \mathrm{d}t \left[ \underbrace{-x_i \sum_{j=i+1}^{n} x_j \alpha_2 (\pi_{j\sigma} - \pi_{i\sigma})}_{s_i \text{ changes to others}} + \underbrace{\sum_{j=1}^{i} x_j x_i \alpha_2 (\pi_{i\sigma} - \pi_{j\sigma})}_{\text{Others change to } s_i} \right] \\ =& x_i(t) + \alpha_1 \mathrm{d}t \left[ \sum_{j=1}^{n} x_j x_i \alpha_2 (\pi_{i\sigma} - \pi_{j\sigma}) \right] \\ \coloneqq& x_i(t) + \alpha_1 \mathrm{d}t \cdot x_i(t) \alpha_2 [\pi_{i\sigma}(t) - \underbrace{\bar{\pi}_{\cdot\sigma}(t)}_{\text{Average } \pi}] \\ \rightarrow \frac{\mathrm{d}}{\mathrm{d}t} x_i(t) =& \alpha_1 \alpha_2 x_i(t) \left[\pi_{i\sigma}(t) - \bar{\pi}_{\cdot\sigma}(t)\right] = \alpha x_i \left[ (\Pi \boldsymbol{x})_k - \boldsymbol{x}^\mathsf{T} \Pi \boldsymbol{x} \right] \qquad \blacksquare \end{aligned}$$

### 2.1.2 Dimension of Strategy State

The last element of $\boldsymbol{x}$, $x_n$, is actually dependent on $x_1, x_2, \cdots, x_{n-1}$. Thus, $\boldsymbol{x} \in \Delta(n)$ only has $n-1$ degrees of freedom. The proof is as follow:

$$\begin{aligned} \because \dot{x}_k &= \alpha x_k \left[ (\Pi \boldsymbol{x})_k - \boldsymbol{x}^\mathsf{T} \Pi \boldsymbol{x} \right] \\ \therefore \dot{x}_n &= \frac{\mathrm{d}}{\mathrm{d}t} \left( 1 - \sum_{k=1}^{n-1} x_k \right) = - \sum_{k=1}^{n-1} \alpha x_k \left[ (\Pi \boldsymbol{x})_k - \boldsymbol{x}^\mathsf{T} \Pi \boldsymbol{x} \right] \\ &= -\alpha \left[ \sum_{k=1}^{n-1} x_k (\Pi \boldsymbol{x})_k - (\boldsymbol{x}^\mathsf{T} \Pi \boldsymbol{x}) \sum_{k=1}^{n-1} x_k \right] \\ &= -\alpha \left[ \boldsymbol{x}^\mathsf{T} \Pi \boldsymbol{x} - x_n (\Pi \boldsymbol{x})_n - (\boldsymbol{x}^\mathsf{T} \Pi \boldsymbol{x})(1 - x_n) \right] \\ &= \alpha x_n \left[ (\Pi \boldsymbol{x})_n - \boldsymbol{x}^\mathsf{T} \Pi \boldsymbol{x} \right] \qquad \blacksquare \end{aligned}$$

Thus, we can see that the last differential equation is actually a linear combination of the previous $n-1$ equations. Our state of dimension $n$ only has $n-1$ degrees of freedom.

## 2.2 Mathematical Prerequisites

### 2.2.1 State-Space Dynamics

Say the mixed strategy (the state of our game) evolves over time subject to the following nonlinear dynamics:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) = \begin{bmatrix} f_1(\boldsymbol{x}(t)) \\ f_2(\boldsymbol{x}(t)) \\ \cdots \\ f_n(\boldsymbol{x}(t)) \end{bmatrix}$$

we can take first order approximation of the system at a *stationary point* $\boldsymbol{a}$ ($\boldsymbol{f}(\boldsymbol{a}) = 0$):

$$\begin{aligned} \dot{\boldsymbol{x}} &\approx \cancel{\boldsymbol{f}(\boldsymbol{a})} + \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}(\boldsymbol{x} - \boldsymbol{a}) \\ \rightarrow \dot{\boldsymbol{x}} = \dot{\boldsymbol{z}} &\overset{\text{1st}}{\sim} A\boldsymbol{z} = A(\boldsymbol{x} - \boldsymbol{a}). \end{aligned}$$

We define a translation of coordinates: $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{a}$. And the term

$$A := \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_1(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f_1(\boldsymbol{x})}{\partial x_n} \\ \frac{\partial f_2(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_2(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f_2(\boldsymbol{x})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_n(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f_n(\boldsymbol{x})}{\partial x_n} \end{bmatrix}$$

is known as the *Jacobian matrix* of the system. It can be viewed as the *state matrix* of the game linearized in the neighborhood of a stationary point of the game.

It is also because of this evolution-like dynamic, the replicator game is also known as evolutionary game theory, applied in the analysis of the evolution of population.

### 2.2.2 Stability

Consider a linear system of the form

$$\dot{\boldsymbol{x}} = A(\boldsymbol{x} - \boldsymbol{a}),$$

we shall define terms to characterize the stability of the system in a neighborhood of the stationary point $\boldsymbol{a}$ by the definiteness, or the eigenvalues of the state matrix:

1. If $A$ is positive definite (positive eigenvalues), it is unstable.
2. If $A$ is negative definite (negative eigenvalues), it is asymptotically stable.
3. If $A$ has eigenvalues of both positive and negative, it is a saddle point.
4. If $A$ has complex eigenvalues, we classify them via the real parts of the eigenvalues. All positive real parts $\to$ unstable. All negative real parts $\to$ asymptotically stable. At least one pure imaginary $\to$ oscillatory.

Such definitions are straightforward, it describes how the magnitude of $\boldsymbol{x}$ evolves as $t$ goes to infinity. And all can be easily determined by simply looking at the phase portrait of the dynamical system.

### 2.2.3 Phase Portrait

Since the system of differential equations

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$$

is *autonomous*, meaning $\boldsymbol{f}$ only depends on the current state of $\boldsymbol{x}$, we can create a phase portrait for it easily.

The phase portrait is a vector field $\boldsymbol{f}$, where at each point $\boldsymbol{x}$ is a vector $\boldsymbol{f}(\boldsymbol{x})$, equivalent to a velocity field for the evolution of the state $\boldsymbol{x}$. We will use the phase portrait for analysis extensively later on.

**Oscillatory Behavior** No oscillatory behavior exists for one-dimensional system dynamics.The proof is omitted.

**Poincaré–Bendixson Theorem** For continuous dynamical systems, chaotic behaviors can only arise if the state space has three or more dimensions.

However, the same does not apply for discrete dynamical systems, where even in dimension of 1, chaotic behaviors can occur. For the purpose of our discussions, in order to not complicate our analysis because of chaotic behaviors, the dynamical system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$$

only has dimension of $\boldsymbol{x}$ up to 2. In terms of action, it means we will have at most 3 actions, where the probability of the last action is dependent of the first two, thus having $\boldsymbol{x}$ with dimension less than 3. But at the same time, we will make the state space to be of 2 dimensional to demonstrate the interesting and non-trivial behaviors.

# 3 Stability Analysis

## 3.1 Nash Equilibrium

For a replicator game in a single population of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}),$$

we define the *stationary points* (or *stationary strategies*) as states $\bar{\boldsymbol{x}} \in \Delta(n)$ such that

$$\boldsymbol{f}(\bar{\boldsymbol{x}}) = 0. \tag{3.1}$$

This means for all $k = 1, 2, \cdots, n$:

$$x_k \neq 0 \Rightarrow (\Pi\bar{\boldsymbol{x}})_k = \bar{\boldsymbol{x}}^\mathsf{T}\Pi\bar{\boldsymbol{x}}.$$

These stationary points includes the mixed Nash equilibria in the original game matrix, since it is identical with the *payoff-equating method.*

For the purpose of our analysis, we define, among the stationary points, the points of *Nash equilibria* as:

**Nash Equilibrium** A strategy/state point $\boldsymbol{x}^N$ of the replicator game $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ is a Nash equilibria only if it is a stationary point ($\boldsymbol{f}(\boldsymbol{x}^N) = 0$) on the phase portrait, and the first order approximation of the system at that point must not be positive definite, i.e.

$$\frac{\partial \boldsymbol{f}(\boldsymbol{x}^N)}{\partial \boldsymbol{x}} \not\succ 0.$$

This definition comes straightly from the observation of the state dynamics.

## 3.2 Rock-Paper-Scissors

First is a classical case of game with oscillatory strategic behavior. This is a rock-paper-scissors game played within a population. At any given moment, the populace are paired up to see who wins the game, and they follow replicator dynamics: if my opponent wins, I might change my action to be the same as his. The game matrix is as follow:

| Opponent / $P$ | Rock $x_1$ | Paper $x_2$ | Scissor $1 - x_1 - x_2$ |
|---|---|---|---|
| Rock $x_1$ | 0 | -1 | a |
| Paper $x_2$ | a | 0 | -1 |
| Scissor $1 - x_1 - x_2$ | -1 | a | 0 |

The matrix above is the profit for the player $P$, and $a > 0$ is a constant payoff for winning. Thus we have:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ 1 - x_1 - x_2 \end{bmatrix} \in \Delta(3),\ \Pi = \begin{bmatrix} 0 & -1 & a \\ a & 0 & -1 \\ -1 & a & 0 \end{bmatrix}$$

From (2.2), we have the game dynamics (taking the proportionality constant $\alpha$ as 1):

$$\begin{cases} \dot{x_1} = x_1 \left[a + (1 - 2a)x_1 - 2ax_2 + (a-1)(x_1^2 + x_2^2 + x_1 x_2)\right] \\ \dot{x_2} = x_2 \left[-1 + 2x_1 + (2-a)x_2 + (a-1)(x_1^2 + x_2^2 + x_1 x_2)\right] \end{cases}$$

This game has 4 stationary points regardless of the value of $a$: $(x_1, x_2) = (0,0)$, $(1,0)$, $(0,1)$, $(1/3, 1/3)$, the last one is of our interest. The associated first order approximation game matrix is:

$$A\left(\frac{1}{3}, \frac{1}{3}\right) = \begin{bmatrix} -\frac{1}{3}a & -\frac{1}{3}(a+1) \\ \frac{1}{3}(a+1) & \frac{1}{3} \end{bmatrix}$$

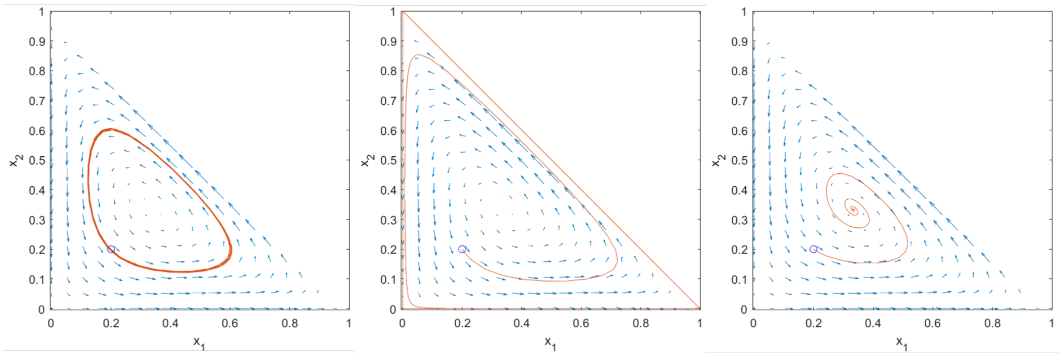
The eigenvalues are complex conjugates for all $a$, and have real parts equal to

$$\frac{1-a}{6}$$

Thus, we can conclude that:

$$\begin{cases} a > 1 & \Rightarrow \text{Asymptotically Stable} \\ a = 1 & \Rightarrow \text{Oscillatory} \\ a < 1 & \Rightarrow \text{Unstable} \end{cases}$$

The plots below (drawn by MATLAB, code provided in appendix), from left to right, have values of $a$ =1, 0.5, and 2 respectively.




The blue circle represents the initial state of $(x_1, x_2) = (0.2, 0.2)$. Oddly enough, the change in the value of winning profit greatly changes how the strategy distribution in population evolves over time.

For $a = 1$, this is in fact a zero-sum game, and the population exhibits stable periodic orbits.

However, for $a < 1$, the strategy of the population switches between rock, paper, and scissors almost purely. This can be understood as: the players have a negative net profit, so one is expected to lose profit playing this game. The best strategy will be then, to reach a tie every time. Thus, all player exhibits the same action. And when a single player changes, say, from rock to paper, all the other players will quickly see that they are losing profit, and will change to playing paper immediately. This is my interpretation of the abrupt transitions.

Lastly, for $a > 1$, the players have a net positive profit. And the best strategy for the whole population is to play an evenly distributed mixed strategy. Though one might not obtain the most payoff in this distribution, he is guaranteed to receive a positive expected payoff.

## 3.3 Two-Population Game

**Game Settings Revisited** Besides utilizing replicator dynamics for a game in a population, where at any given moment, players were paired up in pairs of two to play the given game, it can also be extended. As we will see later on, one can extend it into a two-population game. In fact, we view the whole population as a "single player" that plays a mixed strategy.

**Two-Population Replicator Dynamics** Consider playing a game between two population, having the same action set for each player, and each population has their own strategy. At each moment, players from one population are paired with a player from the other population. Both having the same game matrix $\Pi$ (i.e. the game is symmetric). Let the strategy state of the two populations be $\boldsymbol{x}$ and $\boldsymbol{y}$ ($\in \Delta(n)$) respectively. Then, they can also follow the replicator dynamics of:

$$\begin{cases} \dot{x}_k = \alpha x_k \left[(\Pi \boldsymbol{y})_k - \boldsymbol{x}^\mathsf{T} \Pi \boldsymbol{y}\right] \\ \dot{y}_k = \beta y_k \left[(\Pi \boldsymbol{x})_k - \boldsymbol{y}^\mathsf{T} \Pi \boldsymbol{x}\right] \end{cases} \quad (k = 1, 2, \cdots, n-1).$$

The state dynamics is stated without derivation, since it doesn't exist (note how $k \neq n$)! A way to view it is by taking the population as whole a single player playing mixed strategy $\boldsymbol{x}$. However, an example of its application is as below, included is a detailed procedure on how to draw the phase portrait by hand.

### 3.3.1 Game of Chicken

Imagine in 1962, during the Cuba missile crisis, in the middle of the US-USSR cold war. The two global superpowers are on the verge of launching their nuclear arsenals. In front of the leaders of the two countries, two decisions are present: launch the nukes or don't. We can write the game matrix as:

| US \ USSR | Launch $x_2$ | Don't $1-x_2$ |
|---|---|---|
| Launch $x_1$ | -5,-5 | 3,-3 |
| Don't $1-x_1$ | -3,3 | 0,0 |

$$\Pi = \begin{bmatrix} -5 & 3 \\ -3 & 0 \end{bmatrix}, \text{ Population 1} = \begin{bmatrix} x_1 \\ 1-x_1 \end{bmatrix}, \text{ Population 2} = \begin{bmatrix} x_2 \\ 1-x_2 \end{bmatrix}$$

The dynamic for the game above is:

$$\begin{cases} \dot{x}_1 & = \alpha x_1 \left\{ \left[-5x_2 + 3(1-x_2)\right] - \left[-5x_1x_2 + 3x_1(1-x_2) - 3(1-x_1)x_2\right] \right\} \\ & = \alpha x_1 \left(3 - 3x_1 - 5x_2 + 5x_1x_2\right) = \alpha x_1(1-x_1)(3-5x_2) \\ \dot{x}_2 & = \beta x_2(1-x_2)(3-5x_1) \end{cases}$$

Let the state of the game at a given time $t$ be $\boldsymbol{x} = [x_1, x_2]^\mathsf{T}$, thus enabling us to draw a phase portrait. We can write its dynamic as:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}),\ \boldsymbol{f}(\boldsymbol{x}) \overset{\text{1st}}{\sim} \begin{bmatrix} \alpha(1-2x_1)(3-5x_2) & -5\alpha x_1(1-x_1) \\ -5\beta x_2(1-x_2) & \beta(1-2x_2)(3-5x_1) \end{bmatrix} := A$$

Note that $\boldsymbol{x} \notin \Delta(2)$, but it is a combined state of the two differential equations above. Five stationary points can be found, each with a linearized matrix and eigenvalue-eigenvector pairs shown below:

1. $\bar{\boldsymbol{x}} = [0,0]^\mathsf{T}$

$$A = \begin{bmatrix} 3\alpha & 0 \\ 0 & 3\beta \end{bmatrix} \to \lambda_1 = 3\alpha,\ \lambda_2 = 3\beta,\ \boldsymbol{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ \boldsymbol{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix};$$

2. $\bar{\boldsymbol{x}} = [0,1]^\mathsf{T}$

$$A = \begin{bmatrix} -2\alpha & 0 \\ 0 & -3\beta \end{bmatrix} \to \lambda_1 = -2\alpha,\ \lambda_2 = -3\beta,\ \boldsymbol{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ \boldsymbol{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix};$$

3. $\bar{\boldsymbol{x}} = [1,0]^\mathsf{T}$

$$A = \begin{bmatrix} -3\alpha & 0 \\ 0 & -2\beta \end{bmatrix} \to \lambda_1 = -3\alpha,\ \lambda_2 = -2\beta,\ \boldsymbol{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ \boldsymbol{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix};$$

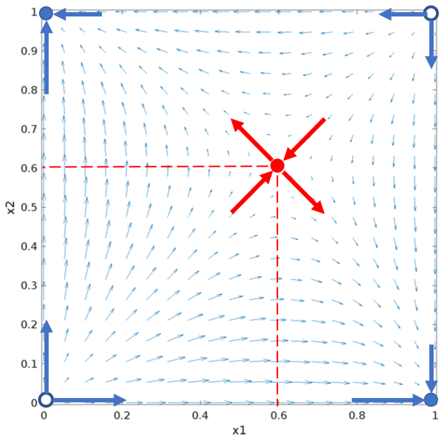
4. $\bar{\boldsymbol{x}} = [1,1]^\mathsf{T}$

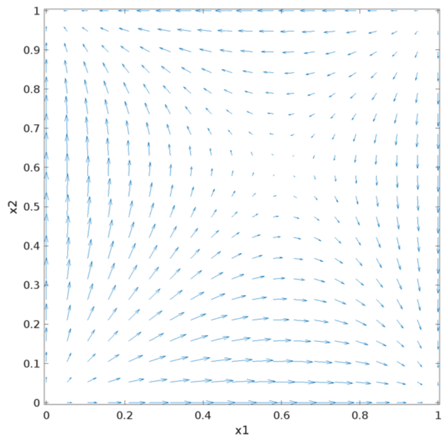
$$A = \begin{bmatrix} 2\alpha & 0 \\ 0 & 2\beta \end{bmatrix} \to \lambda_1 = 2\alpha,\ \lambda_2 = 2\beta,\ \boldsymbol{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ \boldsymbol{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix};$$

5. $\bar{\boldsymbol{x}} = [0.6, 0.6]^\mathsf{T}$

$$A = \begin{bmatrix} 0 & -\frac{6}{5}\alpha \\ -\frac{6}{5}\beta & 0 \end{bmatrix} \to \lambda_1 = \frac{6}{5}\sqrt{\alpha\beta},\ \lambda_2 = -\frac{6}{5}\sqrt{\alpha\beta},\ \boldsymbol{v}_1 = \begin{bmatrix} \sqrt{\alpha} \\ -\sqrt{\beta} \end{bmatrix},\ \boldsymbol{v}_2 = \begin{bmatrix} \sqrt{\alpha} \\ \sqrt{\beta} \end{bmatrix}$$

The phase diagram of the system above can then be easily graphed as follow, where we let $\alpha = 1 = \beta$.





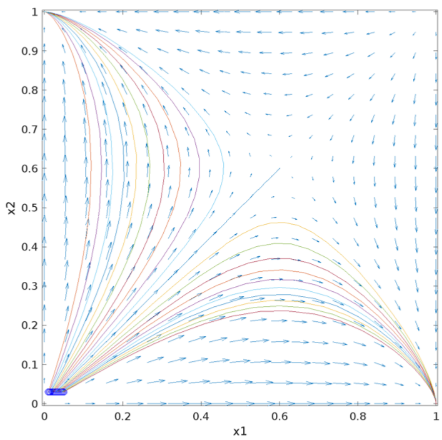

The first and fourth stationary point are dominated by their neighboring points, they represent the points of both launching the missiles and no launches. Moreover, the second and third point are the pure Nash equilibrium of the original single stage game $\Pi$, thus, they are asymptotically stable.

Of most interest for us is the "mixed Nash equilibrium" of $\boldsymbol{x}^N = [0.6, 0.6]^\mathsf{T}$, it is in fact the mixed Nash equilibrium of the original single stage game $\Pi$, and can be readily shown by the payoff-equating method!

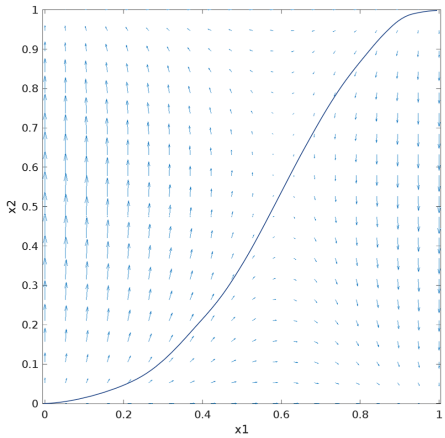
$$\begin{cases} \pi_{\text{US,L}} = -5x_2 + 3(1-x_2) = -3x_2 + 0(1-x_2) = \pi_{\text{US,D}} \\ \pi_{\text{USSR,L}} = -5x_1 + 3(1-x_1) = -3x_1 + 0(1-x_1) = \pi_{\text{USSR,D}} \end{cases}$$

$$\rightarrow x_2 = \frac{3}{5} = x_1$$

However, this is a saddle point, which is to say the linearized matrix of the system dynamics is neither positive (semi)definite nor negative (semi)definite.



For the graph above, the blue circles represents starting states, they are all of close proximity. However, we see that as times evolves, the states diverges into three different regions. This shows the sensitivity of the initial condition for this game. The cold war itself must have states located on the diagonal from $(0, 0)$ to $(1, 1)$. If any of the two nations have a stronger stance than the other, the initial condition will locate at one of the two halves, and the whole game will tend to one nation being destroyed, breaking the equilibrium of cold war. Moreover, if we set $\alpha = 1$, $\beta = 5$, the phase portrait will be:




On the curve going through the diagram are the initial conditions where the two nations will play the mixed Nash equilibrium. If the proportionality constant at the front is different, say $\alpha < \beta$, meaning the US ($x_1$) has a higher tendency of changing his strategy. Also, the area on its half of the plane (the right half) is smaller, meaning less initial states are possible to make the US launch his nuclear arsenal, the US is easier to chicken off.

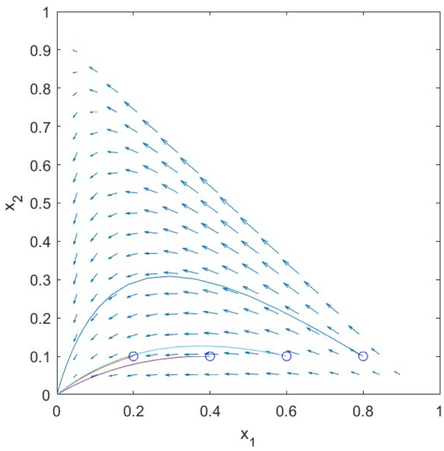
## 3.4 Equilibrium of Single and Two Population Game

Single-population and two-population games are essentially different, but the strategic equilibria reached have a similar meaning. Let us first analyze what the equilibrium means for single-population game, then we shall extend it to two-population game.

**Different Types of Equilibrium** Below are the phase portrait of some single-population game with different game matrix. Again, the different curves represents different evolution of the state based on different initial condition.

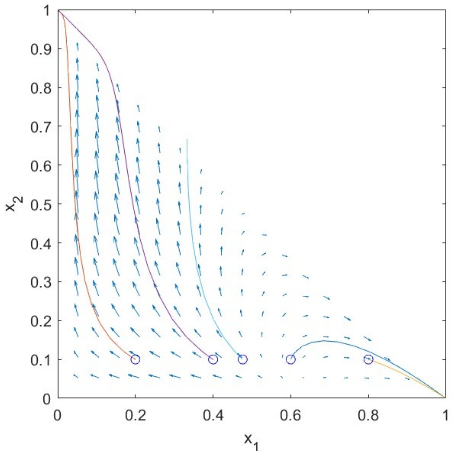
1. Single Dominant Nash

$$\Pi = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 0 & 0 \\ 3 & 2 & 1 \end{bmatrix}$$



2. Two Pure Nash and One Mixed Nash

$$\Pi = \begin{bmatrix} 2 & 4 & 4 \\ 0 & 5 & 10 \\ 0 & 0 & 8 \end{bmatrix}$$



We can see that:

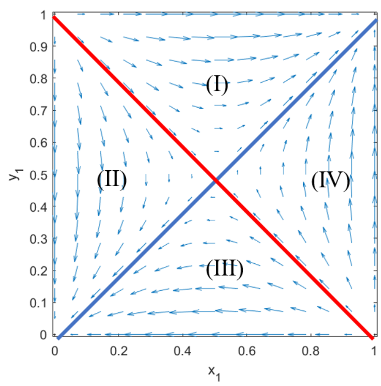
- If a pure Nash equilibrium exists, the system will converge to it with probability one given a random initial condition (see example 1 above).
- If multiple pure Nash exists, so does a mixed Nash (see example 2 above). The initial conditions that are able to reach the mixed strategy (so-called mixed-Nash-reachable states) serve as a boundary between the pure Nash equilibria, and are reached with probability zero. Id est, the mixed Nash is a saddle point, where a small nudge at it would result in evolution towards the pure Nash.
- If no pure Nash exists but a single mixed Nash (like the rock-paper-scissors game), it exhibits oscillatory behaviors. With the system being converging to an equilibrium or oscillates in an orbit periodically determined by the individual payoffs.

**Meaning of Equilibrium to Population and Players** For single population game, each player's strategy is to obtain a higher payoff by replicating others. Thus, the whole populace will reach a *maximal* net payoff at the equilibrium.

The term "maximal" indicates that this is the optimal solution *reachable* and *stable* given an initial condition and the replicating nature of all players. Reachablility of an equilibrium is determined by whether the phase portrait is separated by mixed-Nash-reachable states, and whether the initial condition is on the same side as the equilibrium.

Though the individual players cannot obtain the maximum payoff in the equilibrium state, due to the nature of this dynamic equilibrium, all players should have an equal expected payoff. One will, thus, still be guaranteed a maximal average payoff. In fact, for other reachable strategy points, though at a given instant, one might seem to obtained the maximum payoff. But again, due to the dynamical nature of the game, one loses more on average.

As for two-population games, if there exists a pure Nash, then the results are similar, the payoff of the population state reflects the expected payoff of its players. One possible scenario is that the two populations work together to reach the single pure Nash of max payoff. However, for our case of two-population $2 \times 2$ matrix game, the only other nontrivial settings is where there exists two pure Nash and a mixed Nash. The general phase portrait is as below:




Two curves separate the plane into four quadrants. The red line is converging to the mixed Nash, and the blue diverges to the two pure Nash equilibria. We can tell that the two populations are competing. And the final equilibrium depends on which of the two population is more "aggressive" at the beginning. Under replicator dynamics, the more passive population is like a learning player in a game with an opponent choosing only a specific choice, he will opt to reduce his own loss given the circumstances.

# 4 Perturbations

We introduce the *Gibbs entropy* for a state $\boldsymbol{x} = [x_1 \; x_2 \; \cdots \; x_n]^\mathsf{T} \in \Delta(n)$ as:

$$v(\boldsymbol{x}) = -\sum_{i=1}^{n} x_i \ln x_i \tag{4.1}$$

where $\ln(\cdot)$ represents the natural logarithm. It is demonstrated in [2] and [5] that the perturbed dynamics of

$$\dot{x_k}(t) = \alpha x_k \Big[(\Pi \boldsymbol{x})_k - \boldsymbol{x}^\mathsf{T} \Pi \boldsymbol{x} - \varepsilon(t) \cdot v(\boldsymbol{x})\Big] \tag{4.2}$$

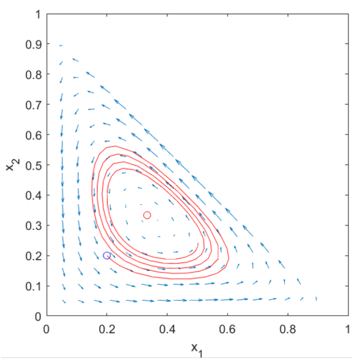
converges to the mixed strategy, given that $\varepsilon(t)$ is a suitable positive decreasing function.

## 4.1 Effects of Perturbations

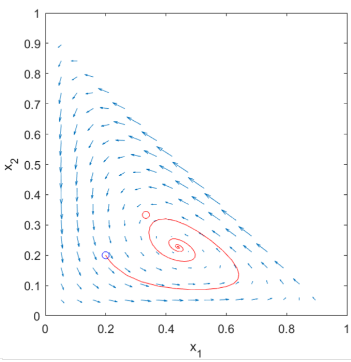
Let us consider the rock-paper-scissors game with $a = 1$, which has an oscillatory mixed Nash equilibrium at $(x_1, x_2) = (1/3, 1/3)$. The perturbed game dynamics is:

$$\Pi = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}, \; \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ 1 - x_1 - x_2 \end{bmatrix}$$
$$\rightarrow \begin{cases} \dot{x_1} = x_1 \left[(\Pi \boldsymbol{x})_1 - \boldsymbol{x}^\mathsf{T} \Pi \boldsymbol{x} - \varepsilon \cdot v(\boldsymbol{x})\right] \\ \\ \dot{x_2} = x_2 \left[(\Pi \boldsymbol{x})_2 - \boldsymbol{x}^\mathsf{T} \Pi \boldsymbol{x} - \varepsilon \cdot v(\boldsymbol{x})\right] \end{cases} \tag{4.3}$$

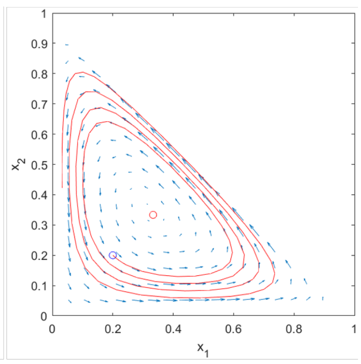
Below are the phase portraits of the perturbed system with different $\varepsilon$:



$\varepsilon = 0.01$



$\varepsilon = 0.1$



$\varepsilon = -0.01$

Again, the blue circle represents the initial condition of the dynamics, and the red circle is the original mixed Nash equilibrium of the game.

A few characteristics can be observed from the figures above:

- The larger the perturbation ($\varepsilon \gg 0$), the faster the convergence to a stationary point.
- The perturbation displaces the Nash equilibrium.
- If $\varepsilon < 0$, the system dynamics diverges.

The meaning of them are discussed in the next subsection.

## 4.2 Meaning of Perturbation

The following is my attempt to explain the validity of the inclusion of perturbation to payoffs.

### 4.2.1 The Gibbs Entropy is the Evaluation of Unexpectedness

For a player, the probability of facing an opponent playing pure strategy $s_i$ is $x_i$. Thus, the smaller $x_i$ is, the less likely it is to meet an opponent playing $s_i$. However, we can quantify the *surprise* of meeting an unexpected opponent as $\mathcal{S}(x_i)$, the quantification must satisfy:

- Expectedness: $\mathcal{S}(1) = 0$, meeting an expected opponent is not a surprise.
- Unexpectedness: $\mathcal{S}(0) = \infty$, meeting an opponent you absolutely sure was impossible is very surprising.
- Additivity: $\mathcal{S}(x_i \cdot x_j) = \mathcal{S}(x_i) + \mathcal{S}(x_j)$, the surprise of multiple independent events happening at the same time is equal to the sum of the surprises of each event.
- Continuity: $\mathcal{S}(\cdot) \in \mathcal{C}$, events with similar probability of happening should lead to similar surprise.

A "natural" choice that matches all the above criteria is the logarithmic function:

$$\mathcal{S}(x) = -\ln x \tag{4.4}$$

The expected surprise for our case of discrete strategy game is thus:

$$\mathbb{E}_S\left[\mathcal{S}\right] = -\sum_{i=1}^{n} x_i \ln x_i \equiv v(\boldsymbol{x}) \tag{4.5}$$

which is, in fact, the Gibbs entropy! Therefore:

*The Gibbs entropy is the expected surprise, measuring the uncertainty and unexpectedness of meeting an opponent.*

### 4.2.2 Perturbation in Payoff

The perturbation in payoff is a counter-measure to the unexpectedness of opponents, i.e., a risk aversion strategy.

For positive $\varepsilon$, the higher the value of unexpectedness, the higher probability it is for one to change strategies; and vice versa for negative $\varepsilon$. Thus, $\varepsilon$ measures how volatile one is– to change strategy easily, lest not begin able to adopt to the stochastic nature of the game. And we see that $\varepsilon$ in (4.2) must be positive for the dynamics to converge to the mixed Nash equilibrium.

With these in mind, we can explain the results obtained in the perturbed rock-paper-scissors dynamics of (4.3), and extend the results to other games.

**Information** The perturbation term is fully expressed through the strategy of the opposing population. Thus, it is only applicable if all players have ”the strategy distribution of the opposing population” as an information. The observability of such information is also worth discussing– if not observable, one can create an observer to provide a state identification, but this is not the main topic for this study.

**Reaction to Unexpectedness** Since $\boldsymbol{x}$ (population 1) faces off with $\boldsymbol{y}$ (population 2): players in population 1 reacts to the unexpectedness of strategy distribution in population 2. Therefore, the perturbed game dynamics is

$$\dot{x_1} = x_1\left[(\cdots) - \varepsilon \cdot v(\boldsymbol{y})\right],$$

instead of

$$\dot{x_1} = x_1\left[(\cdots) - \varepsilon \cdot v(\boldsymbol{x})\right].$$

And vice versa for dynamics of population 2. The term $\varepsilon$ represents the reaction of the players of a population to unexpectedness. For $\varepsilon > 0$, it means one might choose to change action at next stage of game, even if he is winning at current stage. As for $\varepsilon < 0$, it means one has a higher probability of sticking to his current action at next stage even when he is currently losing.

As for replicator games within a single population, the game dynamics will be

$$\dot{x_k}(t) = \alpha x_k \left[ (\Pi \boldsymbol{x})_k - \boldsymbol{x}^\mathsf{T} \Pi \boldsymbol{x} - \varepsilon(t) \cdot v(\boldsymbol{x}) \right],$$

as presented in (4.2), since one's opponent is in the same population as oneself.

**Displacement and Convergence** For $\varepsilon > 0$, the game converges to a mixed strategy, yet it differs from the unperturbed mixed Nash equilibrium. This can easily be verified since

$$\boldsymbol{x}^N = \left[1/3, 1/3, 1/3\right] \rightarrow v(\boldsymbol{x}^N) \neq 0 \rightarrow \dot{\boldsymbol{x}}(\boldsymbol{x}^N) \neq 0.$$

Therefore, two solutions are feasible. The first would be to modify $v$ such that $v(\boldsymbol{x}^N) = 0$, another suitable solution would be to have a time-varying $\varepsilon = \varepsilon(t)$.

Thus, convergence and displacement of equilibrium are competing with each other. One must balance between them to acquire desired results.

**Divergence of Strategy** If instead, $\varepsilon < 0$, it means one is less likely to change when facing high uncertainties within the opponent population. And the state of the system shall "diverge" to a pure Nash equilibrium.

**Inconsistency** Note that for the above analysis, we do not include any proof from the decision of a single player to the dynamics of the entire population. The reason is that it doesn't exist, and one can tell by simply checking that the dynamics of $x_n$ is inconsistent with the others. The next section shall provide a better perturbation term: it considers the individual action of each player, and is consistent.

### 4.2.3 Reaction to Unexpectedness

Here we provide an interpretation on the effect of reaction to unexpectedness has on the state of the whole population.

Three cases are possible: $\varepsilon = 0$, $\varepsilon > 0$, and $\varepsilon < 0$.

- The first case is where each player cares nothing about unexpectedness, and change his strategy solely on the difference in payoffs.
- The second case is where players are more prone to change strategy. Not only does one change strategy based on difference in payoffs, but he

also changes strategy easily when faced with an unexpected opponent, no matter win or lose. The effect of such strategy is that in the end, the state of the population will converge to a dynamical equilibrium–a mixed Nash equilibrium.

- The third case is the opposite. Though one still changes strategy because of the difference in payoff, creating oscillating orbits, he is more likely to stick to his original strategy in the next round if faced with an unexpected opponent. This dynamic will cause the state of the system to converge to a pure Nash equilibrium. But with the unexpected nature of the game, it'll oscillate between the pure Nash equilibria.

These results, in fact, are quite interesting. Mathematically, the converging/diverging dynamics can be explained as follow: Consider a nonuniform distribution, say $x_1 > x_2$, and the total perturbation $v(\boldsymbol{x})$ is large. For $\varepsilon > 0$, it'll contribute to the dynamics as

$$\dot{x}_k \propto x_k \left[(\cdots) - \varepsilon \cdot v(\boldsymbol{x})\right] \approx -\varepsilon \cdot x_k v(\boldsymbol{x})$$

Thus, for $x_1$, it decreases faster, and $x_2$ decreases slower. And equilibrium is struck when they are about uniformly distributed. And when $\varepsilon < 0$, the bigger of the two increases faster, thus creating the diverging dynamics.

## 4.3 Different Choices of Perturbation

The perturbation function need not be the Gibbs entropy function. In fact, using $v(x)$ results in problems such as displacement of equilibrium and $\dot{x}_n$ being not dependent. Shown below is one of the alternatives.

Similar as before, in a single population game: for a single player in the population, we shall modify his strategy-changing probability function $x_{i,j}$ of (2.3) as

$$x_{i,j}(t) = \begin{cases} \alpha_2 \left[\pi_{j\sigma}(t) - \pi_{i\sigma}(t)\right] + \varepsilon \cdot \left(-\ln x_j\right) & , \pi_{i\sigma}(t) < \pi_{j\sigma}(t) \\ \varepsilon \cdot \left(-\ln x_j\right) & , \pi_{i\sigma}(t) \geq \pi_{j\sigma}(t) \end{cases} . \tag{4.6}$$

The change in strategy depends on two factor: the difference in payoff between one and his opponent, his reaction to facing an unexpected opponent.

As mentioned before, $-\ln x_j$ represents the surprise of meeting an opponent with action $s_j$. A positive $\varepsilon$ means one is more likely to change strategy, even if he is winning in the current round; and one is more likely to stick to his choices for $\varepsilon < 0$. Then:

$$\dot{x}_k = \alpha_1 x_k \left[\alpha_2 \cdot (\pi_{i\sigma} - \bar{\pi}_{\cdot\sigma}) - \varepsilon \cdot \left(v(\boldsymbol{x}) + \ln x_k\right)\right] \tag{4.7}$$

$$\rightarrow \dot{x}_k = \tilde{\alpha} x_k \left[(\Pi \boldsymbol{x})_k - \boldsymbol{x}^\mathsf{T} \Pi \boldsymbol{x} - \tilde{\varepsilon} \cdot \left(v(\boldsymbol{x}) + \ln x_k\right)\right] \tag{4.8}$$

And since this perturbation term is derived from the action of the individual players in the population, it satisfies the following theorem:

**One Reduced Degree of Freedom** Similar to the proof provided prior, this new dynamics has only $n-1$ independent differential equations for $\boldsymbol{x} \in \Delta(n)$. Unlike the dynamics of (4.2), which will run into problems when trying to proof the dependence of $\dot{x}_n$ on other states.

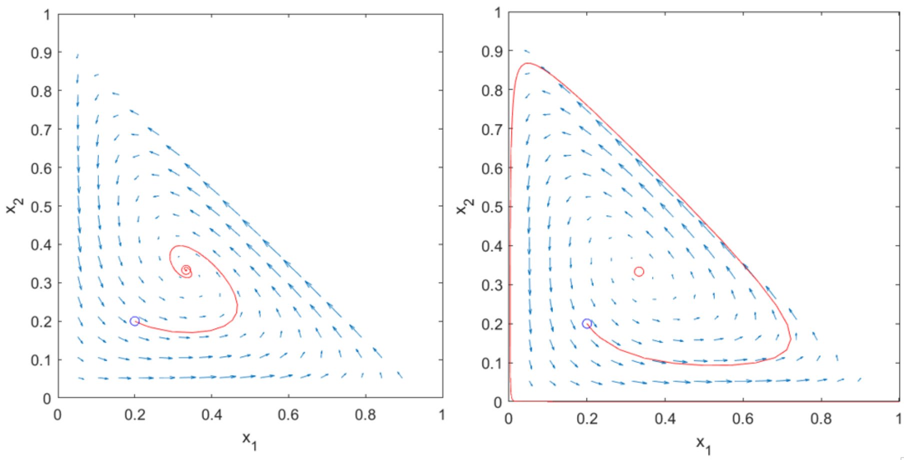
**Convergence at Uniform Distribution** Moreover, this new perturbation also results in convergence to a displaced equilibrium. To be exact, this new dynamics will converge to $x_1 = x_2 = \cdots = x_n := x_{\text{eq}} = 1/n$ for positive $\varepsilon$, since:

$$v(\boldsymbol{x}_{\text{eq}}) + \ln x_{\text{eq}} = -n \cdot \left(\frac{1}{n} \ln \frac{1}{n}\right) + \ln \frac{1}{n} = 0.$$

Which makes it the perfect perturbation for our rock-paper-scissors game (if convergence to the mixed strategy of $(x_1, x_2) = (1/3, 1/3)$ is intended that is).

## 4.4 Examples of New Perturbation Function

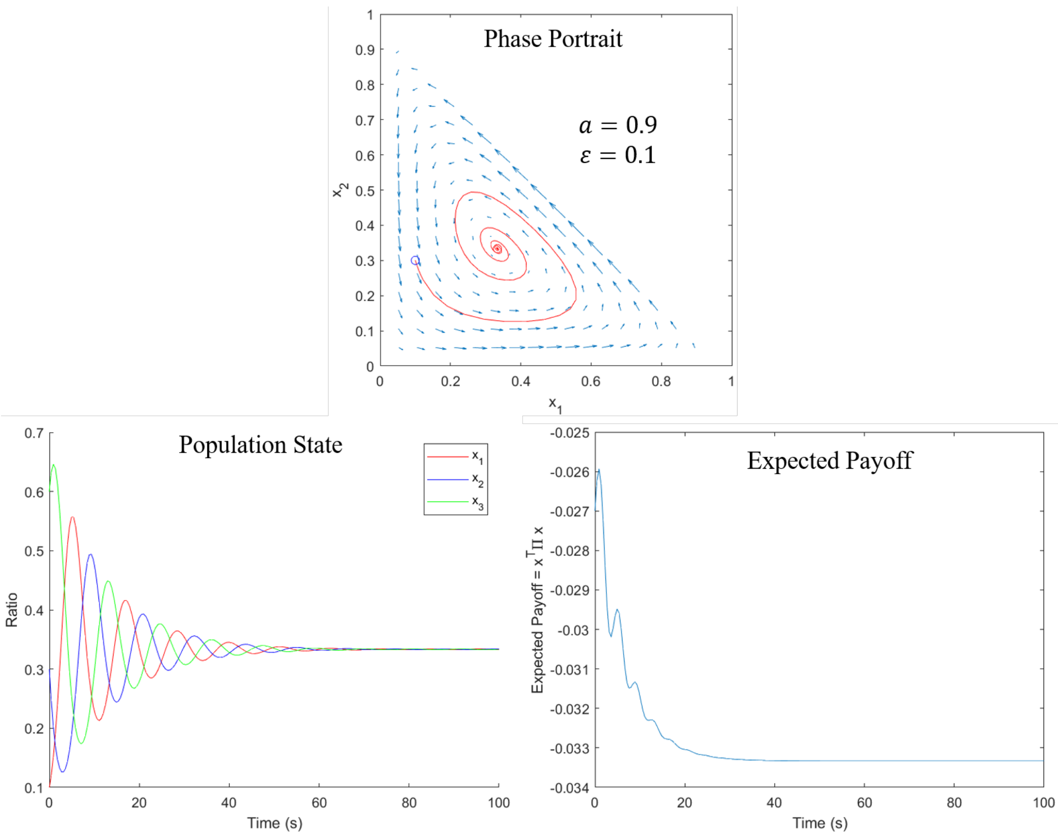
For this new perturbed dynamics of the rock-paper-scissors game: take $a = 1$ and the perturbation as $v(\boldsymbol{x}) + \ln x_k$, the results are:




$\varepsilon = 0.1$ $\varepsilon = -0.1$

where the red circles above represents the original Nash equilibrium of $(x_1, x_2) = (1/3, 1/3)$. The effects of the winning payoff $a$ and $\varepsilon$ compete with each other. With appropriate manipulation of $\varepsilon$, one can force the game to converge/diverge for an arbitrary choice of $a$ and initial distribution of strategies.

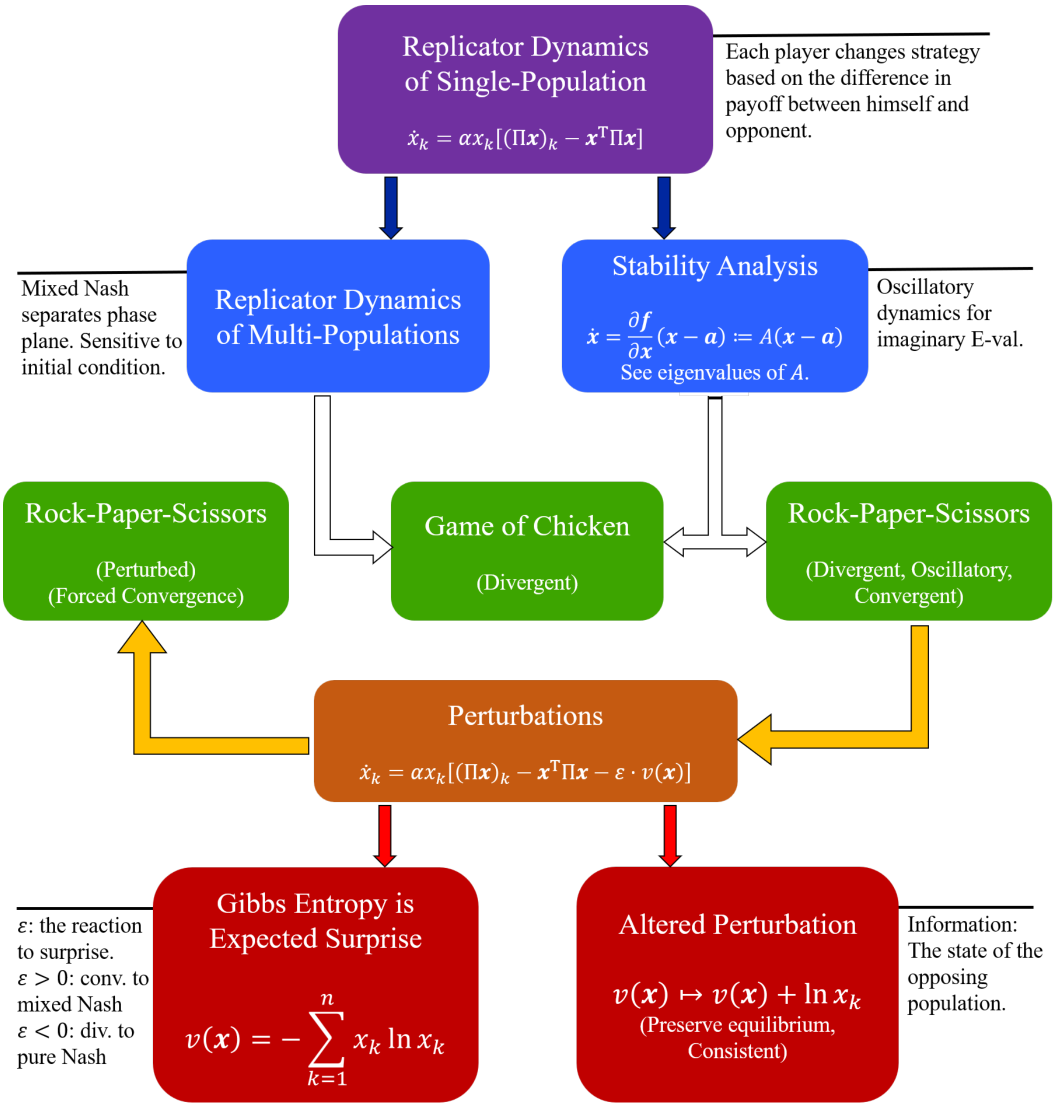
An example is as follow:




For $a < 1$, the state is suppose to evolve to a pure Nash equilibrium, but due to the inclusion of the perturbation, the game converges to the mixed Nash. Moreover, it seems that the strategy distribution converges exponentially under the perturbation of equation (4.8). However, the equilibrium reached is not one with higher profit (obviously), but instead, just a stable and attracting one.

# 5 Results

The results can be summarized as the graph below.




In the future, more can be discussed upon several topics listed below:

1. The observability of state used in perturbation.
2. Convergence of state under perturbation.
3. Better formulation of two-population replicator dynamics.
4. Asymmetric game matrix of two-population game.

# References

[1] Jun Wang, *Learning in Repeated Games*, Computer Science, UCL, Oct 15 2018, SJTU.

[2] Justin Kang, *A Review of Smooth Fictitious Play and Fictitious Play in Repeated Games*, Department of Electrical and Computer Engineering, University of Toronto. https://justinkang221.github.io/files/paper5.pdf, June 10, 2021.

[3] Tim Rees, *An Introduction to Evolutionary Game Theory*, UBC Department of Computer Science.

[4] *The Replicator Dynamic (Draft)*, https://www.ma.imperial.ac.uk/ svanstri/GamesAndDynamics/The%20Replicator%20Dynamic%20(Draft).pdf.

[5] Lucas Baudin, and Rida Laraki, *Smooth Fictitious Play in Stochastic Games with Perturbed Payoffs and Unknown Transitions*, Université Paris-Dauphine - PSL, arXiv:2207.03109v1 [cs.GT], July 7, 2022.

[6] Julien Perolat et al., *From Poincaré Recurrence to Convergence in Imperfect Information Games: Finding Equilibrium via Regularization*, February 19, 2020.

# Appendix

## A. MATLAB Code for Plotting Phase Portrait

The code for the rock-paper-scissors game with perturbation referenced in subsection 4.4 is shown below (ran on MATLAB R2021b):

```
%% Game Setup %%
a = 1; % winning payoff
P = [ 0,-1, a;
      a, 0,-1;
     -1, a, 0]; % game matrix

%% Perturbation term $v(\boldsymbol{x})$ %%
v1 = @(X) (-X(1)*log(X(1))-X(2)*log(X(2))-X(3)*log(X(3)));
v2 = @(X,k) (v1(X)+log(X(k)));
EPS = 0.1; % reaction to surprise $\varepsilon$

%% Game Dynamics
vec = @(X) [X(1);X(2);1-X(1)-X(2)]; % state
e = @(k) unit(k); % unit vector
g = @(t,X,k) (X(k)*((e(k)'*P*X)-(X'*P*X)-EPS*(v2(X,k)))); % $\dot{x}_k$
f = @(t,X) [g(t,vec(X),1);g(t,vec(X),2)]; % $f(\boldsymbol{x}) = [\dot{x}_1, \dot{x}_2]$
%------------------------------------------------------------------------------%
%% Phase Portrait %%
x1 = linspace(0,1,20);
x2 = linspace(0,1,20);
[x,y] = meshgrid(x1,x2);

u = zeros(size(x));
v = zeros(size(x));
t = 0;
for i = 1:numel(x)
    if x(i)+y(i)<1
        Xdot = f(t, [x(i),y(i)]);
    else
        Xdot = [0;0];
    end
    u(i) = Xdot(1);
    v(i) = Xdot(2);
end
%% Settings of Portrait
quiver(x,y,u,v);
set(gcf,'color','w');
xlabel("x_1");
ylabel("x_2");
axis equal tight;
```

```
%% Analytic Solution given Initial State %%
hold on
    plot(1/3,1/3,"ro") % equilibrium point
    % initial state = [1/5,1/5]
    [ts, ys] = ode45(f, [0,50], [1/5,1/5]);
    plot(ys(:,1),ys(:,2),'r'); % evolution curve
    plot(ys(1,1),ys(1,2),'bo'); % starting point
hold off

%% Functions %%
% Unit vectors
function vec = unit(k)
    vec = zeros(3,1);
    vec(k) = 1;
end
```

Similarly, the code for two-population game is obtained by modifying the "Game Setup", "Game Dynamics", and "Functions" sections. Take the game of chicken from subsection (3.3.1) as an example.

```
%% Game Setup
P = [-5,3;-3,0];
```

```
%% Game Dynamics
vec = @(X) [X;1-X]; % vectorize
e = @(k) unit(k); % unit vector
g = @(t,X,Y,k) X(k)*((e(k)'*P*Y)-(X'*P*Y));
f = @(t,X) ...
    [g(t,vec(X(1)),vec(X(2)),1);g(t,vec(X(2)),vec(X(1)),1)];
```

```
%% Functions
% Unit vectors
function vec = unit(k)
    vec = zeros(2,1);
    vec(k) = 1;
end
```