

NONLINEAR DYNAMICS AND ENTROPY IN EVOLUTIONARY GAMES

JADEN OLAH, TITUS PARKER, SEIJI YANG

ABSTRACT. Cyclic interactions appear in many places throughout nature. Competition, mobility, and mutually cooperative strategies can lead to ‘cyclic domination’, where species’ mutual competition leads to an cyclic evolutionary stable strategy. In this report, we review foundational literature and concepts in nonlinear dynamical systems theory and chaos theory. We discuss examples of chaotic behavior arising in various systems. We motivate and use bifurcation theory to study these systems, starting from a first principles exploration of bifurcations before showing how they can be used to characterize chaotic systems. In addition, we use bifurcations to study the cyclic dominance of evolutionary games, reviewing literature covering cyclic dominance in both well-mixed systems and systems with spatial mobility. We then introduce a new definition of regional entropy which we use to investigate the behavior of a simple evolutionary game.

1. INTRODUCTION

Nonlinear dynamical systems and chaos theory provides a mathematical framework to understand evolutionary games. We focus on mathematical evolutionary games which involve cyclic dominance, a naturally occurring phenomena. We will provide the needed mathematical background to use tools such as bifurcation theory to analyze different types of Rock-Paper-Scissors (RPS) dynamics. In particular, we will see how we can mathematically model competition, reproduction, and mobility.

This report begins by building up the mathematical foundations in section 2. We introduce differential equations, fixed points, stability, phase portraits, and bifurcation theory. We cover the three main bifurcations, saddle-node, transcritical, and pitchfork. We then discuss linearization, the Jacobian matrix, limit cycles, and Hopf bifurcations. We then talk about attractors and chaotic systems, namely the Lorenz equations and the discrete logistic map.

We then discuss a paper by *Szolnoki et al.*[8] in section 3. The paper discusses evolutionary game theory in the context of cyclic dominance. We examine various transition probability parameters such as dominance-removal, dominance-replacement, reproduction, and mutation. We discuss some first order dynamics and intuitively build up a system of differential equations to model this RPS game. We then expand our discussion to spatial mobility, reaction–diffusion formulations, and the emergence of spiral wave patterns. We end the section with discussion of the Complex Ginzburg-Landau equation.

This report is submitted as part of the capstone requirement for the Mathematics major at Stanford University.

The final portion, section 4, of this paper pertains the GPU simulations of our own evolutionary game. We describe our GPU-accelerated simulation framework using CuPy, the rules of the RPS model, and the normalization of interaction rates. We introduce our entropy metric that describes spatial complexity of species. We also include the pseudocode of our algorithm to allow readers an intuitive understanding of the underlying mechanisms of the game. We then move onto results, including population trajectories, and plot the relationship between mobility and our measure of entropy. We observe and interpret the impact mobility has on pattern complexity and its relation to theoretical expectations based off of bifurcation analysis.

2. BACKGROUND AND LITERATURE REVIEW

2.1. Differential Equations. A *differential* equation is one that relates an unknown function to its derivative. Differential equations arise in many real-world contexts, often where the derivative is with respect to time. Differential equations often model phenomena that change over time.

For example, the following equation models exponential growth/decay, something widely seen in natural sciences, like in radioactive decay of an isotope.

$$\frac{dN}{dt} = kN(t) \quad (2.1.1)$$

where N could be the amount of some chemical isotope for example, and k models the rate of growth/decay. If $k < 0$, then the isotope *decays exponentially*, and if $k > 0$ then the isotope *grows exponentially*. Note here how the function of interest, N , is defined explicitly in terms of its derivative.

2.1.1. Solving Differential Equations. We say a differential equation *has an explicit solution* if we can find an explicit equation for the function independent of its derivative. For equation (2.1.1), a clever guess will yield a solution of

$$N = e^{kt}. \quad (2.1.2)$$

Instead of a clever guess, however, we can solve it algebraically:

$$\begin{aligned} \frac{dN}{dt} &= kN(t) \\ \frac{1}{N(t)} dN &= kdt \\ \int \frac{1}{N(t)} dN &= \int kdt \\ \ln(N(t)) &= k \\ N(t) &= e^{kt}. \end{aligned}$$

This method is called *separation of variables*, and only works when an equation is *separable*. Unfortunately, not all differential equations are as easily solved. In fact, it is rare in sciences that differential equations can ever be *analytically solved* (meaning there exists an explicit algebraic solution): for example, the Schrödinger equation, which models quantum phenomena, is only analytically solvable with the hydrogen atom. When we add more electrons to our atom, the interactions amongst electrons leads to *non-separability*, leading to a lack of an analytic solution.

There are other causes of unsolvability of differential equations. Note in equation (2.1.1), our right hand side does not depend *explicitly* on time – it only depends on time through $N(t)$. This is called *autonomy* in a differential equation, where the equation does not depend on the independent variable (here being time). Non-autonomous differential equations are harder to solve. Similarly, non-linear equations, where for a given equation $(d/dt)x(t) = \dot{x} = f(x(t))$ has f a *non-linear* function of $x(t)$, are often unable to be analytically solved. In these cases, the equations must be approximated with *numerical methods*, which are algorithms that provide reasonable approximations of the solution to a given level of accuracy.

2.2. Stable Points. We know that differential equations are used to model many real world phenomena, from population of bacteria in a petri dish, to chemical reactions, to quantum wavefunctions. However, in many of these examples, the object we’re modeling reaches some stable state – the number of bacteria in a petri dish does not grow forever, or the whole world would be bacteria! This concept is called *stability*, and it can be translated to differential equations with the notion of *stationary solutions*. Many terms are used to describe the same thing (equilibrium points, fixed points, stationary points, stable solutions), but in this report, we stick to *fixed point* or *stationary solution*.

For an example, consider the following, called the *logistic growth equation*:

$$\frac{dN}{dt} = rN \left(1 - \frac{N}{k}\right) \quad (2.2.1)$$

where N is the population of a species, and r, k are both parameters. In fact, most terms of (2.2.1) equation have nice physical meaning - k is called the *carrying capacity* of a species, and $(dN/dt) = \dot{N}$, as the *rate of change of the population* is otherwise known as the population growth. For a population modeled in this way, intuitively, the population is stable if it is neither growing nor shrinking. Thus, we want to find solutions where $\dot{N} = 0$. Solving (2.2.1) as a quadratic yields $N = k$ or $N = 0$. This nicely introduces the following:

Definition 2.2.1. A *fixed point* or stationary solution of a differential equation $\dot{x} = f(x)$ is a function $x^*(t) = c$ that satisfies the differential equation, where c is some constant, not necessarily in one dimension. This yields $\dot{x}^*(c) = 0$.

First, note in the above definition, x and c can both be vectors. This definition, then, generalizes to differential equations in *many dimensions*. For a three dimensional system in 3D Euclidean space, then, a fixed point would satisfy both the differential equation, and $(\dot{x}, \dot{y}, \dot{z}) = (0, 0, 0)$. Finally, we typically depict a fixed point with an asterisk, such as x^* .

2.3. Phase Portraits. What if we can’t solve a differential equation, and we don’t have access to a computer? Armed with stationary solutions, fixed points, and basic algebra, we can visualize behavior of differential equations in ways that give us more information. We demonstrate this with the *phase portraits* of a two-dimensional system.

For this subsection, consider a two-dimensional system of differential equations defined as follows:

$$\frac{dx}{dt} = \dot{x} = a_{11}x + a_{12}y, \quad \frac{dy}{dt} = \dot{y} = a_{21}x + a_{22}y.$$

It is easy to see that this is a *linear* system, and we can write it as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$$

where

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}.$$

Because this system has two dimensions, we can easily plot it in two spatial dimensions (the plane). Thus, it is a *planar system*, and every solution to this differential equation, is some curve in the plane, parameterized by time. This curve in the plane is called an *orbit* or *trajectory*.

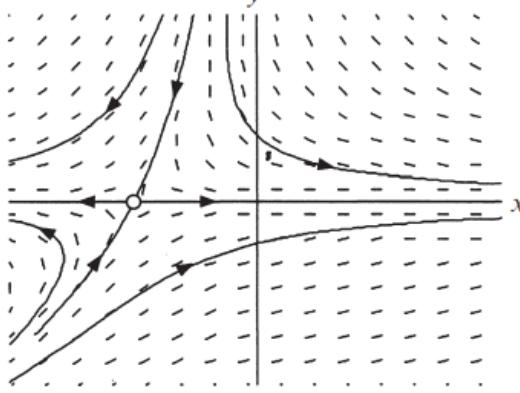


FIGURE 2.3.1. A phase portrait of an arbitrary system [7].

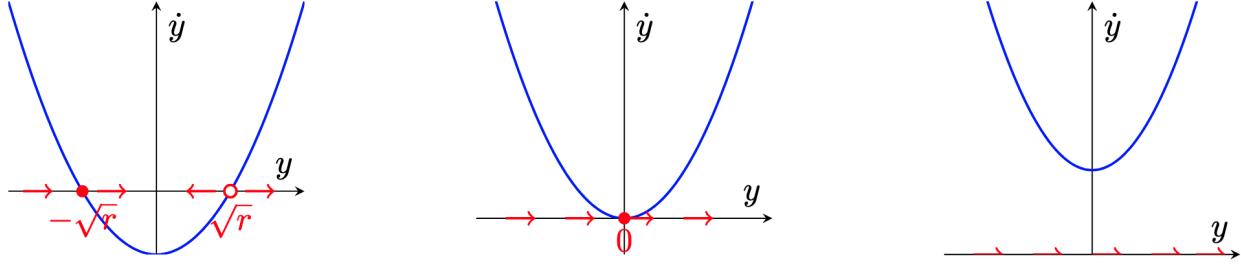
What happens when we plot these orbits on the plane? It depends on the system, but in many cases, plotting planar systems lets us easily visualize stable points. In the above example, trajectories are given with little line segments, with specific trajectories drawn with solid lines. We can see that in *phase space*, the space of dependent variables (here the plane), trajectories either shoot off to $(\pm\infty, 0)$, or converge to the fixed point in negative x-space.

2.4. Bifurcations. A *bifurcation* captures the behavior of a fixed point appearing/disappearing when parameters change. There are three main classes of bifurcations in one dimension: saddle-node bifurcations, transcritical bifurcations and pitchfork bifurcations.

2.4.1. Saddle-Node Bifurcations. *Saddle-node bifurcations* describe the process of fixed points being created and destroyed. Consider the following equation:

$$\frac{dx}{dt} = rx^2.$$

The above is called the *algebraic normal form* of a saddle-node bifurcation the general algebraic formulation of all systems that exhibit saddle-node bifurcations. For negative values of r , we have a stable point and an unstable point, at $x = -\sqrt{-r}$ and $x = \sqrt{-r}$, respectively. As we increase the value of r , these fixed points become closer and closer to one another, until they merge into one fixed point that is neither stable nor unstable (but semi-stable) when $r = 0$. As we further increase r to some positive value, these fixed points disappear.

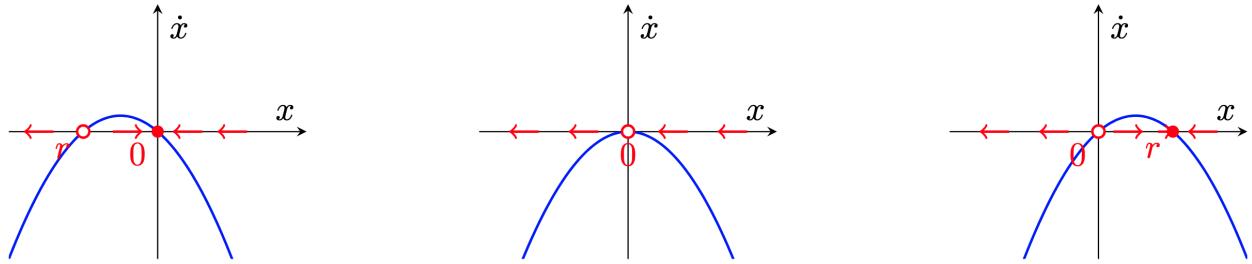
FIGURE 2.4.1. Demonstration of a Saddle-Node Bifurcation as parameter r changes.

2.4.2. Transcritical bifurcations. Unlike saddle-node bifurcations, *transcritical bifurcations* do not involve the spontaneous appearance/disappearance of a fixed point under some parameter's variation. Instead, a fixed point undergoes a *change* in stability, moving from stable to unstable or vice versa.

For an example, consider a one-dimensional differential equation

$$\frac{dx}{dt} = rx - x^2.$$

A fixed point in one dimension is where $\frac{dx}{dt} = 0$, which implies that fixed points exist at $x = 0$ and $x = r$. However, once we move r to 0 from the left, we see that, although the fixed point at $x = 0$ is initially stable, it becomes unstable as $r > 0$, with the fixed point at $x = r > 0$ now stable. This is captured in the bifurcation diagrams below, where the fixed point at 0 changes stability.

FIGURE 2.4.2. Demonstration of a Transcritical Bifurcation as parameter r changes.

2.4.3. Pitchfork bifurcations. *Pitchfork bifurcations* often arise in physical systems displaying some kind of symmetry phenomenon. In these cases, the fixed points appear and disappear in symmetrical pairs. The algebraic normal form of a Pitchfork bifurcation is given by

$$\dot{x} = rx - x^3.$$

Observe that if $y = -x$, then $\dot{y} = ry - y^3 = r(-x) - (-x)^3 = -rx + x^3 = rx - x^3$. The cubic term of the normal form enforces this symmetry and so the bifurcation is invariant under the change of variables.

When $r \leq 0$, there exists one stable fixed point, however, from 2.4.3 we see that in the $r = 0$ case (center image) that the decay is much slower than the negative r case (leftmost image). This

is known as the *critical slowing down* phenomenon. The pitchfork behavior from which these bifurcations derive their name is readily apparent in the following figure.

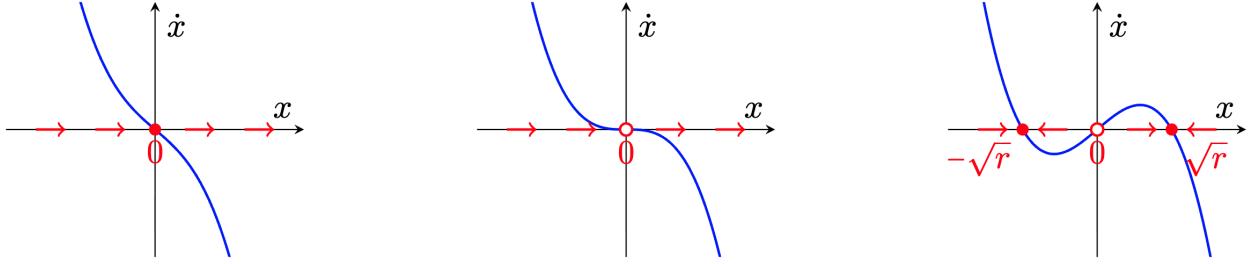


FIGURE 2.4.3. Demonstration of a Pitchfork Bifurcation as parameter r changes.

In figure 2.4.3, we can see the way the fixed points are eliminated by varying the value of r . Note finally that, while these examples are in one dimension, bifurcations occur in many dimensions, and \dot{x} can be generalized to a vector $\dot{\mathbf{x}}$.

2.5. Phase Plane Analysis for Second Order Systems. For many nonlinear differential equations, it is difficult to find analytic solutions. So often we try to analyze the qualitative behavior of such systems. The most salient features for us to analyze are:

- (1) Fixed points (i.e. where $f(x^*) = 0$).
- (2) Closed orbits, where for all t , there exists a T such that $x(t+T) = x(t)$.
- (3) The arrangement of the solution trajectories near fixed points and closed orbits.
- (4) The stability of fixed points and closed orbits.

The first major result that we present here is a theorem to establish the existence and uniqueness of such solution trajectories.

Theorem 2.5.1. (Existence and Uniqueness Theorem) Suppose $f \in C^\infty[D]$, i.e., f is a function that is continuous and infinitely differentiable over some domain D where $D \subset \mathbb{R}^N$ and consider the initial value problem, $\dot{x} = f(x)$ where $x(0) = x_0$. Then, for $x_0 \in D$, the initial value problem has a solution $x(t)$ over some time interval $(-\tau, \tau)$ which is unique.

Observe that the solution is not necessarily extant and unique over the whole space, but only over some interval $(-\tau, \tau)$. An important implication of this theorem is that *solution trajectories never intersect in phase space*. This is because if they were to, then the point of their intersection would correspond to a point shared by two solutions, which violates Theorem 2.5.1. This is why the theorem is also sometimes known as the No Crossing Theorem.

2.6. Linearization. Perturbation theory is the idea of approximating solutions using a power series for a solution to a simpler version of the problem. The first few terms encode most of the information, and higher order terms account for small adjustments. This also has use in finding *numerical* solutions of differential equations. Linearization is a special case of perturbation theory. It is the process of replacing a nonlinear system with a linear system that best describes it. Consider the following 2D system

$$(\dot{x}, \dot{y}) = (f(x, y), g(x, y)) \quad (2.6.1)$$

with fixed point at (x^*, y^*) . Now we wish to analyze the behavior of a trajectory an arbitrarily small perturbation away from the fixed point. To do this, let

$$(u, v) = (x - x^*, y - y^*). \quad (2.6.2)$$

Now, re-writing the original system in terms of (u, v) yields

$$\begin{aligned} \dot{u} &= \dot{x} = f(x^* + u, y^* + v) \\ &= f(x^*, y^*) + uf_x + vf_y + O(u^2, v^2, uv) \\ &= uf_x + vf_y + O(u^2, v^2, uv). \end{aligned}$$

Where O refers to a function of the parameters, $O(u^2, v^2, uv)$ where O is used to approximate additional terms in the equation by their behavior as the input parameters go to infinity.

Similarly for v we find that

$$\dot{v} = ug_x + vg_y + O(u^2, v^2, uv).$$

Now, we can re-organize the system into the form

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} f_x & f_y \\ g_x & g_y \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + O(u^2, v^2, uv). \quad (2.6.3)$$

The matrix

$$\begin{pmatrix} f_x & f_y \\ g_x & g_y \end{pmatrix} \quad (2.6.4)$$

is known as the *Jacobian matrix* for the system, where importantly, it is only evaluated at the known fixed points. The eigenvalues λ_1 and λ_2 determine its behavior. These eigenvalues are complex conjugates (see Math 53 Textbook [4] for details). We denote the real part as $\text{Re}(\lambda)$ and the imaginary part as $\text{Im}(\lambda)$. The signs of $\text{Re}(\lambda)$ and $\text{Im}(\lambda)$ determine the behavior (as illustrated in Table 2.6.1).

$\text{Re}(\lambda)$	$\text{Im}(\lambda)$	Type	Behavior
< 0	$= 0$	Stable node	move directly toward the fixed point
> 0	$= 0$	Unstable node	move directly away from the fixed point
< 0	$\neq 0$	Stable spiral	spiral inward; damped oscillations
> 0	$\neq 0$	Unstable spiral	spiral outward; growing oscillations
$= 0$	$\neq 0$	Center	closed orbits; neutrally stable periodic motion
one < 0 , one > 0	any	Saddle	approach along one direction and repel along another

TABLE 2.6.1. Table of system behavior corresponding to different eigenvalue types.

This provides a powerful general approach to understanding the behavior of second order systems. We can identify the fixed points and compute the Jacobian of the system, and evaluate it at the fixed points. Specifically, we can evaluate the eigenvalues of the Jacobian to classify fixed points,

providing information about local stability. The behavior of the system can then be broken up as follows:

- (1) *Repellents* (also known as sinks). This occurs when both eigenvalues of the Jacobian have positive real parts.
- (2) *Attractors* (also known as sources). This occurs when both eigenvalues of the Jacobian have negative real parts.
- (3) *Centers*. This occurs when both eigenvalues are pure imaginary.

Figure 2.6.2 provides a good illustration of all three phenomena under the variation of a parameter a .

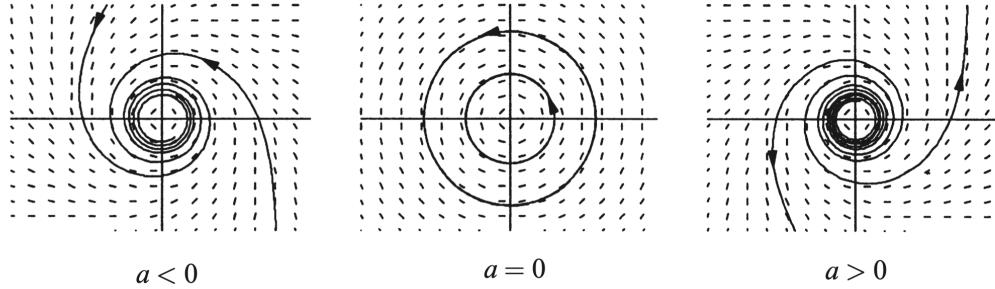


FIGURE 2.6.2. The trajectories of system entering into either attractors (left), Centers (center) or repellents (right) based on the variation of a parameter a [7].

If $\text{Re}(\lambda) \neq 0$ for all λ_i , then the fixed point is said to be hyperbolic. To see this analysis in action we will analyze a classical physical system: a swinging pendulum. We will neglect damping and external drivers, so that the pendulum is an ideal oscillator, with governing dynamical equation

$$\frac{d\theta^2}{dt^2} + \frac{g \sin(\theta)}{L} = 0. \quad (2.6.5)$$

Where θ is the angle of the pendulum from the downward vertical, L is the length and g is the force of gravity. First, we will non-dimensionalize the equation by introducing a dimensionless time parameter $\tau = \omega t$ for frequency $\omega = \sqrt{g/L}$. Then the equation becomes

$$\ddot{\theta} + \sin(\theta) = 0 \implies \begin{pmatrix} \dot{\theta} \\ \dot{\nu} \end{pmatrix} = \begin{pmatrix} \nu \\ -\sin(\theta) \end{pmatrix} \quad (2.6.6)$$

where ν is the dimensionless angular velocity. Now from this, we can see that the fixed points of this system are $(\theta^*, \nu^*) = (k\pi, 0)$ for some integer k . Therefore, we compute the Jacobian matrix as

$$\begin{pmatrix} 0 & 1 \\ -\cos(\theta) & 0 \end{pmatrix}. \quad (2.6.7)$$

Evaluating this at the fixed point $(0, 0)$ is

$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

so the center is nonlinear and trajectories will orbit around it. Similarly evaluating the system at $(\pm\pi, 0)$ yields

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \Rightarrow \lambda = \pm 1$$

which we know corresponds to a saddle point. We can continue this process and generate a phase portrait for the oscillating pendulum [2].

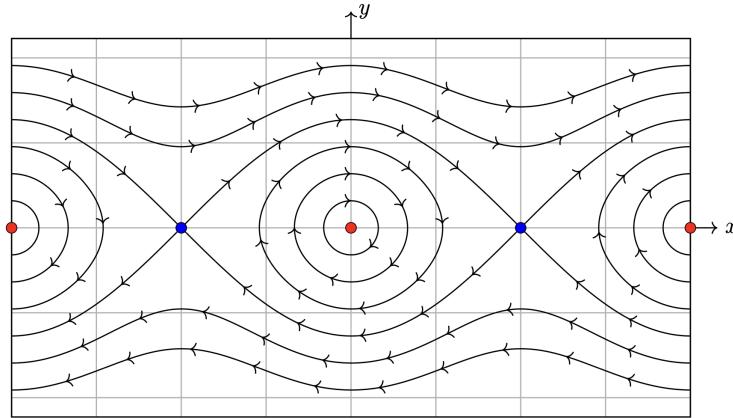


FIGURE 2.6.3. The trajectories for the linearized pendulum system in phase space showing entering into either centers or saddle points [4].

2.7. Limit Cycles. Limit cycles, much like fixed points, describe isolated closed trajectories where trajectories (both outside and inside the closed region) either spiral towards or away from it. Limit cycles appear when a previously stable equilibrium loses stability. This transition will be touched upon later when discussing Hopf bifurcations. The following theorem guarantees the existence of a limit cycle.

Theorem 2.7.1. (*Poincaré–Bendixson Theorem*) If the following holds,

- (1) R is a closed and bounded subset of the plane;
- (2) the system $\dot{x} = f(x)$ is continuously differentiable on an open set containing R ;
- (3) R contains no equilibrium points;
- (4) there exists a trajectory C that remains entirely within R for all future time;

then either:

- C itself is a closed periodic orbit or
- C spirals toward a closed periodic orbit as $t \rightarrow \infty$.

In either case, the region R contains at least one closed orbit.

This theorem has a nice intuitive meaning—if we find a well-behaved region of phase space with no fixed points, within which an orbit is contained, then that region has a limit cycle. This is because our orbit cannot converge to a fixed point, and topologically, so long as it stays within that region, as time goes to infinity, it must be periodic. We do not provide a proof of 2.7.1 due to its difficulty.

2.8. Hopf Bifurcations. Hopf Bifurcations describe the process of limit cycles being created and destroyed. There are two main types of Hopf Bifurcations: supercritical and subcritical. In the table above, the stability of a spiral depends on the sign of $\text{Re}(\lambda)$. A Hopf bifurcation happens exactly where $\text{Re}(\lambda) = 0, \text{Im}(\lambda) \neq 0$. The pair of complex conjugate eigenvalues move across the imaginary axis as the parameters are changed. This point is where a fixed point changes its stability and begins to oscillate. This process can either create or destroy a stable limit cycle. The former, where a stable limit cycle is created in place of the now unstable fixed point is called a supercritical Hopf bifurcation. The latter, where an unstable limit cycle collapses into a fixed point is called a subcritical Hopf bifurcation.

2.8.1. Supercritical Hopf Bifurcation. The parameter μ is the control parameter that shifts the real part of the eigenvalues. As μ crosses zero, the equilibrium changes stability and a limit cycle is born. For the supercritical Hopf bifurcation, the complex conjugate eigenvalues cross the imaginary axis as a parameter μ crosses the critical value μ_c . This indicates that $\text{Re}(\lambda)$ has changed sign from negative to positive, so the spiral loses stability. As $\text{Re}(\lambda)$ approaches zero, the oscillations subdue until they neither grow nor shrink at the bifurcation point. Figure 2.8.1 shows the typical geometry of the supercritical Hopf bifurcation. When $\mu < 0$, the equilibrium is a stable spiral, so trajectories spiral inward as seen on the left. As μ approaches 0, the oscillations decay more slowly until, at $\mu = 0$, a small-amplitude periodic orbit is born. For $\mu > 0$, this limit cycle becomes stable and grows in amplitude, while the equilibrium becomes unstable. The right panel shows trajectories spiraling outward from the now unstable fixed point and being captured by the stable limit cycle.

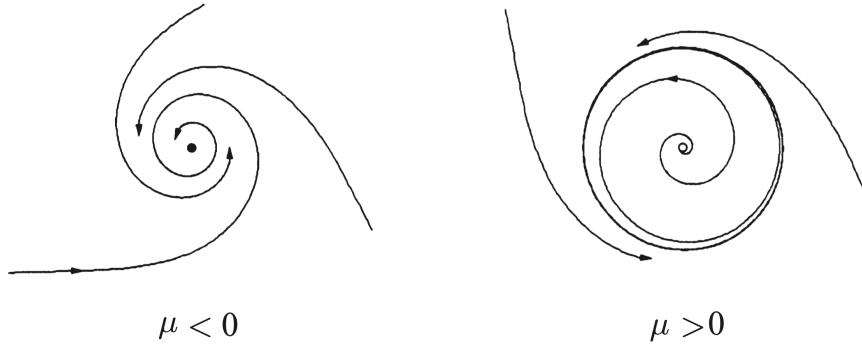


FIGURE 2.8.1. Supercritical Hopf bifurcation behavior as μ is changed [7].

2.8.2. Subcritical Hopf Bifurcation. For the subcritical Hopf bifurcation, the complex conjugate eigenvalues also cross the imaginary axis due to adjusting parameter μ to cross μ_c . However, $\text{Re}(\lambda)$ changes sign from positive to negative, which in turn makes the fixed point stable. Before the bifurcation, there is an unstable limit cycle which shrinks and collapses into a fixed point as μ exceeds μ_c . Trajectories near this point tend to get repelled away from this point. The phase portraits in Figure 2.8.2 illustrate this behavior. For $\mu < 0$, the equilibrium is unstable and is surrounded by an *unstable* limit cycle, shown as the dashed curve in the left panel. Trajectories inside or outside of this repelling cycle are pushed away from it. As μ increases toward 0, the

unstable cycle shrinks and collapses. Once $\mu > 0$, the limit cycle disappears and the equilibrium becomes a stable spiral, as shown in the right panel.

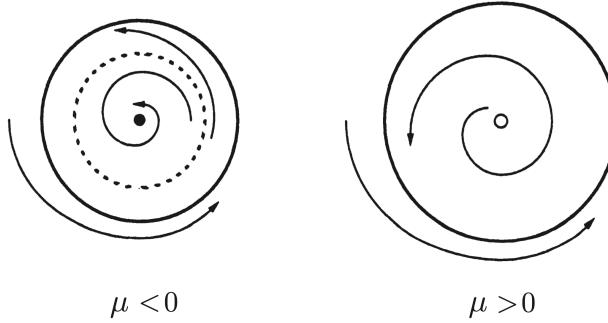


FIGURE 2.8.2. Subcritical Hopf bifurcation behavior as μ is changed [7].

2.9. Chaos. Chaos describes deterministic motion that is highly sensitive to initial conditions, leading to long-term unpredictability. Chaos first appears when the system has three variables. A classic example of such a system are the Lorenz Equations

2.9.1. Lorenz Equations.

$$\begin{cases} \dot{x} = \sigma(y - x), \\ \dot{y} = x(\rho - z) - y, \\ \dot{z} = xy - \beta z, \end{cases} \quad (2.9.1)$$

The Lorenz equations are an example of chaos in continuous time that models heat convection. Here, x , y , and z represent fluid velocities and temperature gradients. σ , ρ , and β are positive parameters. Despite being deterministic, the system has many features of chaos. Specifically, aperiodicity (trajectories never settle to a fixed point or orbit), sensitivity to initial conditions, and bounded complex motion. The butterfly image seen in 2.9.1 is a fractal set called the Lorenz attractor, and is a prototypical example of how chaos may arise from simple nonlinear dynamics.

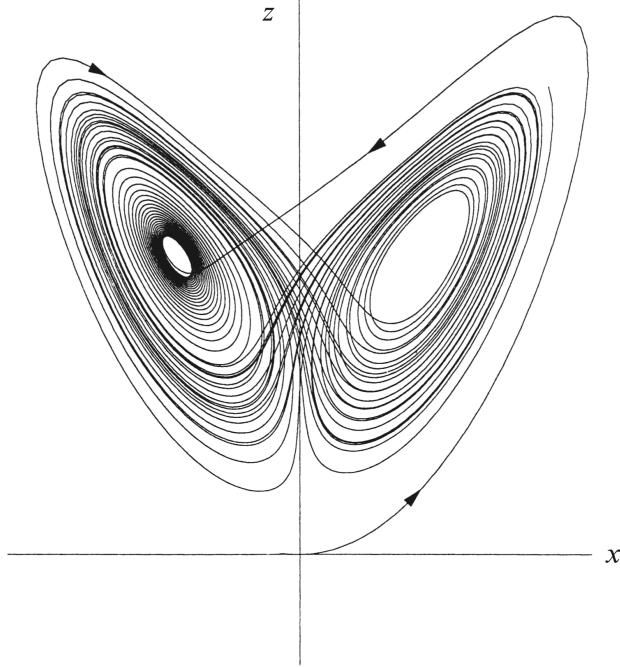


FIGURE 2.9.1. Butterfly pattern when plotting $x(t)$ against $z(t)$ [7].

2.9.2. The Logistic Map. An example of chaos arising from discrete time is the logistic map, which was previously referenced in equation (2.2.1). Let

$$x_{n+1} = rx_n(1 - x_n), \quad 0 < x_n < 1 \quad (2.9.2)$$

where x_n represents the state at time n , and r is a growth parameter. For smaller values of r , the system settles to a stable fixed point, indicating that the system may be quite predictable in the long-term. However, as one increases the value of r to above 3, the number of equilibrium states double. As r is even further increased to above 3.4, it doubles once again. This sequence of period doubling begins to happen more frequently at more rapid intervals as r is increased, eventually leading to chaotic behavior with an exponential number of equilibrium points. Figure 2.9.2 visually demonstrates this chaos by plotting the equilibrium points of the system as a function of r .

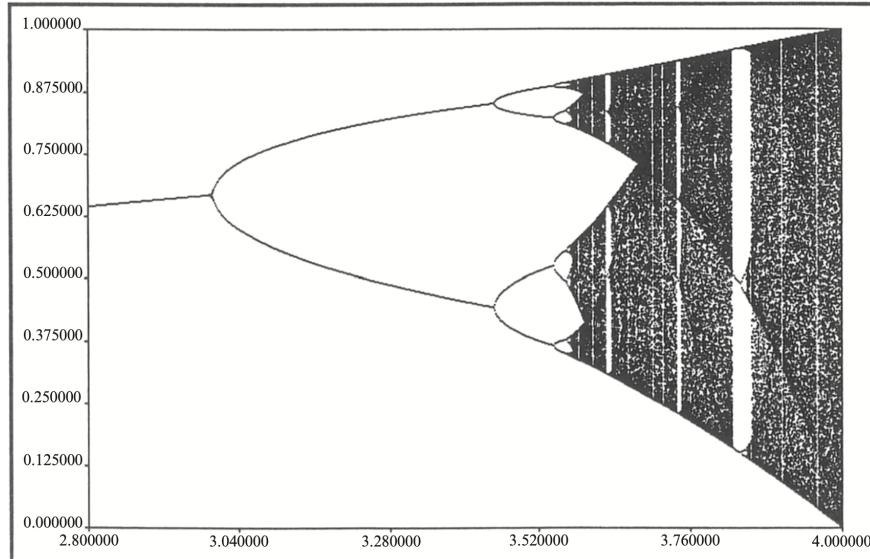


FIGURE 2.9.2. Plot of the period doubling bifurcation with equilibrium points as a function of r [7].

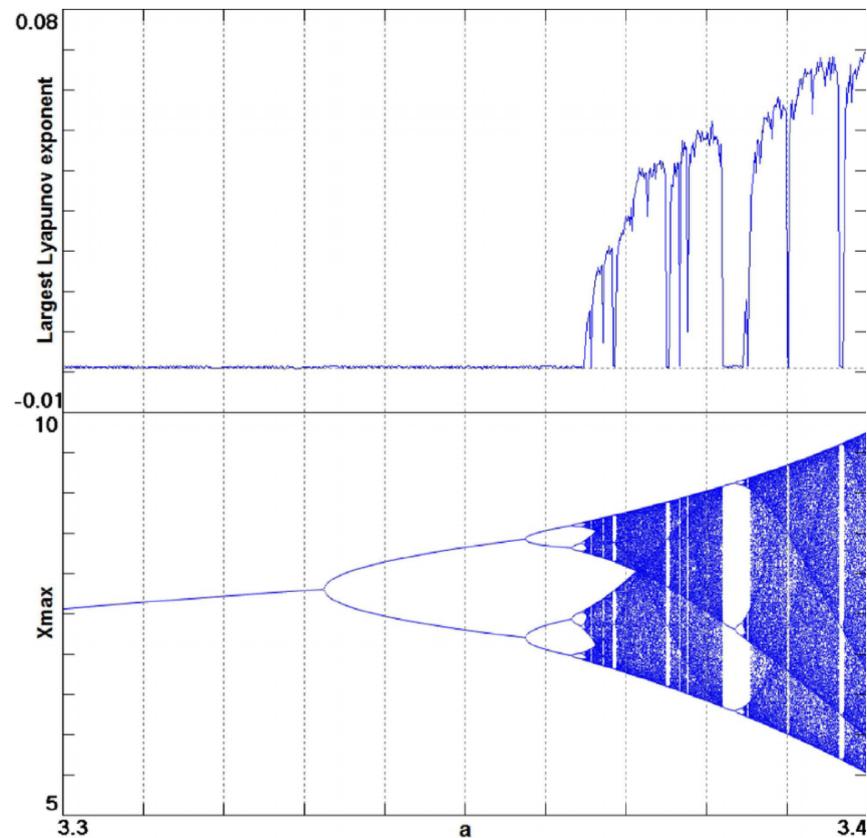


FIGURE 2.9.3. Plot of the period doubling against the Lyapunov exponent [5].

A fascinating feature of the period doubling bifurcation is its connection to the Mandelbrot set. One may notice that amidst the chaos, the system occasionally returns to order (as marked by the clear sections within the graph). These locations of order and predictability correspond to stable regions of the Mandelbrot set. Figure 2.9.4 illustrates this unique connection by overlaying the two systems on top of one another. The same graph with the r values of the discrete logistic map is overlaid with the x -axis of the Mandelbrot set. It is clear that the stable sets of both fractals happen at the same intervals. The rate at which how quickly the period-doubling bifurcations accumulate in the discrete logistic map is related to the Feigenbaum Constant. As the parameter r increases, each successive bifurcation occurs at an interval smaller by a factor approaching $\delta \approx 4.669$, the Feigenbaum constant. This same constant appears in the Mandelbrot set, describing the rate at which new bulbs are attached to the parent cardioid, with a similar doubling pattern. As a result, the familiar period-doubling behavior of the period doubling bifurcation is embedded into the nature of the Mandelbrot set.

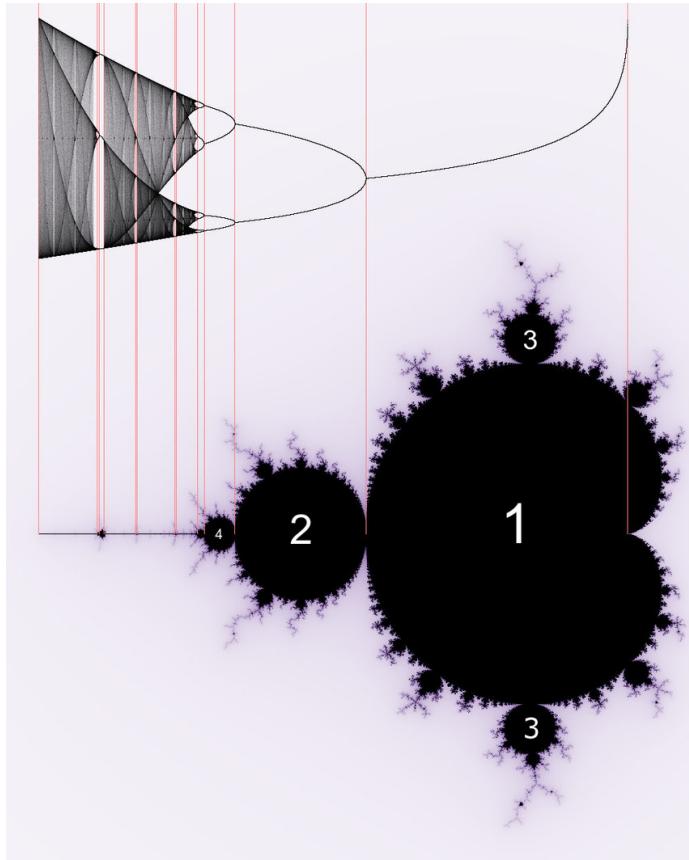


FIGURE 2.9.4. Discrete logistic map overlaid with Mandelbrot set [3].

2.10. Attractors. Recall from section 2.5 that we have loosely defined a fixed point for a system as an *attractor* if both eigenvalues of the Jacobian evaluated at the fixed point have negative real part. However, the introduction of limit cycles suggests that we should extend this definition to have a more topological flavor. Therefore, we introduce the following definition.

Definition 2.10.1. An *attractor* is a closed set A (that is to say, a set that contains all of its limit points) with the following properties:

- 1) If $x(t_0) \in A$ for some initial t_0 then $x(t) \in A$ for all t .
- 2) There exists an open set (a set that for every point p contains all of the points sufficiently near to p) U which contains A such that if $x(t) \in U$ then the limit of the distance from $x(t)$ to A goes to 0 for sufficiently large t , i.e., A attracts all trajectories sufficiently close to it with enough time.
- 3) A is the smallest set possible that satisfies conditions 1 and 2. There does not exist a subset of A that satisfies the previous conditions.

This last condition is significant because even if a set attracts all of the trajectories, it may contain a smaller one that also does so. Only this smaller set can be defined as an attractor. An important observation is that the behavior of trajectories even within attractors can offer some surprising results. Namely, trajectories can exhibit sensitive dependence on initial conditions within the attractor itself. This is where the name *strange attractor* is derived from. The Lorentz attractor in Figure 2.9.1 is the most famous example of a strange attractor.

We can also ask the question of how the attractors change their stability if not only the initial conditions, but also the underlying parameter values are varied. Figure 2.10.1 illustrates this best. Showing the known global stability for $x(t)$ given parameter $r < 1$. At $r = 1$ the origin loses its stability and undergoes a supercritical pitchfork bifurcation into two attracting fixed points. At $r = 24.74$ the fixed points undergo a subcritical Hopf bifurcation and absorb an unstable limit cycle thereby losing stability. The stability of the solution against the changing parameter values is the conceptual idea behind the example of the period doubling bifurcation in section 2.9.1 and the universality analysis and is one of the most powerful tools for characterizing nonlinear systems.

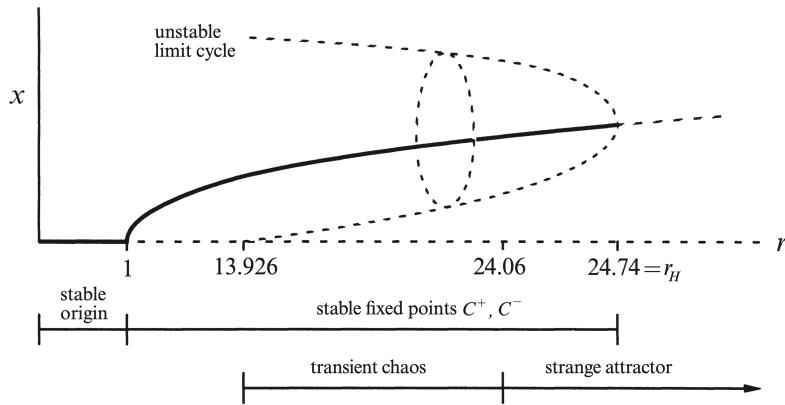


FIGURE 2.10.1. The changing stability of the origin of a Lorenz system with increasing parameter values r . Note the change in stability of the attractors with increasing r [7].

2.11. Generating Unique Pseudo Chaotic Attractors. How can we quantify a chaotic attractor? One way to do so is with a *Lyapunov exponent*. This quantifies the exponential separation between two trajectories under a given system of difference equations. Consider two trajectories perturbed from each other by some small $\delta(0)$. We plot the divergence of these in Figure 2.11.1.

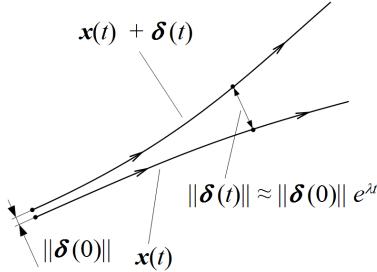


FIGURE 2.11.1. The Lyapunov Divergence.

Here, the trajectories' separation is determined uniquely by $e^{\lambda t}$:

$$\|\delta(t)\| \approx e^{\lambda t} \|\delta(0)\| \quad (2.11.1)$$

From this equation, we can recover the *Lyapunov Exponent* λ , which determines the separation of these two trajectories. It's worth noting that there are as many Lyapunov exponents as there are dimensions in our system. We can use the Lyapunov exponent to characterize other bifurcations that we have already explored such as the period doubling bifurcation. Figure 2.9.3 shows the period doubling bifurcation with the Lyapunov exponent overlain.

With this, we can create a basic algorithm to simulate chaotic-attractor-like images. Recalling that an attractor describes trajectories that (loosely) stay within an open set. Chaos is captured with a positive Lyapunov exponent, implying exponential separation. Therefore, we can search for systems with positive Lyapunov exponents *and* attractor-like behavior. This gives us the following algorithm, which we use to generate images of chaotic attractors.

Algorithm 1 Generating Strange Attractors

Require: T (some integer *after* early transient dynamics have settled)

- 1: Pick random point (x,y)
 - 2: Pick random coefficients a_0, \dots, a_{11}
 - 3: Set $x_\epsilon, y_\epsilon = x \pm \mathcal{N}(\epsilon), y \pm \mathcal{N}(\epsilon)$, at some step in our algorithm.
 - 4: **for** 10000 iterations **do**
 - 5: **if** $|(x, y) - (x', y')| < 10^{-10}$ **then** break (convergence to a point)
 - 6: **end if**
 - 7: **if** $|x'|, |y'| > 10^{10}$ **then** break (divergence)
 - 8: **end if**
 - 9: $\lambda \leftarrow \frac{1}{N-T} \sum_{t=T}^N \log\left(\frac{\delta_t}{\delta_{t-1}}\right)$ (Lyapounov Exponent)
 - 10: If $\lambda > 0$, trajectories diverge exponentially, indicating chaotic behavior.
 - 11: **end for**
 - 12: return points
-

We plot these points as a scatter graph to produce images like the following:

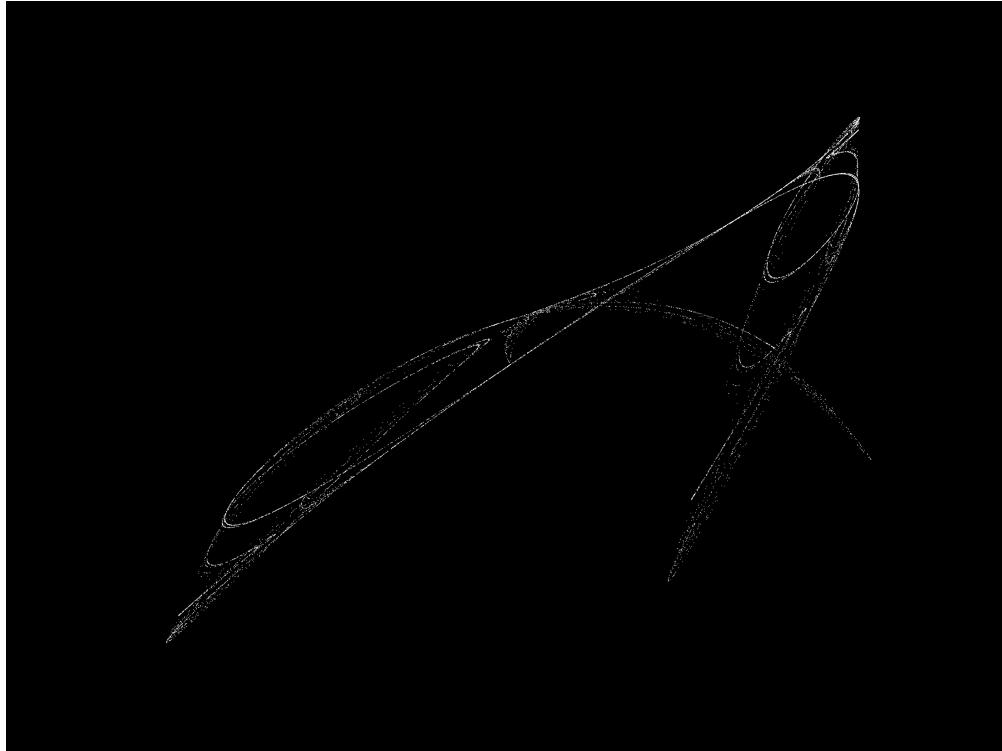


FIGURE 2.11.2. A generated attractor using 2.11

Although the results are visually striking, it is worth noting that this is not a *true* chaotic attractor. This is because our system is in the plane: two dimensional systems cannot be truly chaotic because they would have overlapping trajectories, which implies non-determinism and is impossible under non-stochastic differential equations. We can get away with this in generated images because we are simulating iterated maps, whose discreteness doesn't have the same 'overlapping trajectories' analogue with the continuous case.

3. EVOLUTIONARY GAMES [8]

3.1. Introduction. An evolutionary game is an application of game theory to evolving populations in biology, where multiple ‘strategies’ (or species) face off against each other. Here, we apply tools from bifurcation theory to characterize the rich dynamical behavior of rock-paper-scissors (RPS), formulated as a spatial evolutionary game.

Spatial evolutionary games are evolutionary games where the interaction network is *topologically limited*. Thus, instead of every player/organism interacting with every other player, (which can be modeled as functions over edges of a complete graph), interactions are limited to specific topologies (neighbors of a lattice, edges of a grid, etc). Specifically, we focus on the paper by Szolnoki *et al.*[8] which covers oscillatory behavior in extinction dynamics, structured populations, and spatial constraints with mobility. We see how stable coexistence through spiral-like patterns emerge. The paper also connects these spatial systems to the complex Ginzburg–Landau equation.

3.2. Cyclic Dominance. We focus on rock paper scissors (RPS) behavior, also known as cyclic dominance, in well-mixed populations. A well-mixed population is an environment in which every individual interacts equally with all others. As in the section above, we will name the three species A , B , and C .

We now define a few key constants. The first is the dominance-removal rate p which models cases where the winning species kills or removes the loser. “ \emptyset ” denotes an empty site. Here the winner eliminates the loser, creating a free space.

$$AB \xrightarrow{p} A\emptyset, \quad BC \xrightarrow{p} B\emptyset, \quad CA \xrightarrow{p} C\emptyset, \quad (2.1)$$

Next, we have the dominance-replacement rate z which turns the loser into another copy of itself. This does not change the total number of individuals, it simply changes what species occupies the site without adding or removing the exact number of individuals.

$$AB \xrightarrow{z} AA, \quad BC \xrightarrow{z} BB, \quad CA \xrightarrow{z} CC, \quad (2.2)$$

We also have reproduction rate q into empty sites. this describes the rate at which a species reproduces into an empty site, spreading into free space and increasing the number of individuals of its species.

$$A\emptyset \xrightarrow{q} AA, \quad B\emptyset \xrightarrow{q} BB, \quad C\emptyset \xrightarrow{q} CC, \quad (2.3)$$

Finally, we have a mutation rate μ , which describes the rate at which individuals spontaneously mutate into another species. This can manifest in genetic mutations or microbes switching phenotype. This helps stabilize systems as it reintroduces all species.

$$A \xrightarrow{\mu} \begin{cases} B \\ C \end{cases} \quad B \xrightarrow{\mu} \begin{cases} A \\ C \end{cases} \quad C \xrightarrow{\mu} \begin{cases} A \\ B \end{cases} \quad (2.4)$$

Let $a(t)$, $b(t)$, and $c(t)$ represent the densities of species A , B , C respectively. let $r = \frac{1}{2}(a + b + c)$ be the density of empty sites. The following deterministic system is produced in well-mixed infinite populations where $r_\emptyset = 1 - (a + b + c)$ represents the density of empty sites:

$$\begin{aligned} \frac{da}{dt} &= a [qr_\emptyset + z(b - c) - pc] + \mu(b + c - 2a), \\ \frac{db}{dt} &= b [qr_\emptyset + z(c - a) - pa] + \mu(a + c - 2b), \\ \frac{dc}{dt} &= c [qr_\emptyset + z(a - b) - pb] + \mu(a + b - 2c). \end{aligned} \quad (2.5)$$

The intuitive interpretation for the terms of A is as follows (the intuition generalizes for B and C by replacing the species in accordance with the RPS rules):

- qr_\emptyset represents reproduction of A into an empty space.

- $z(b - c)$ represents the effect of cyclic dominance on A . Since A beats B , an A individual “gains” $+zb$ when it meets a B individual. Similarly, it loses $-zc$ when it meets a C individual since A loses to C .
- $-pc$ represents the loss for A as a result of being killed by C and leaving an empty space.
- $\mu(b+c-2a)$ represents the total net gain and loss from mutation. $b+c$ accounts for species B and C mutating into A and $-2a$ accounts for species A mutating into either B or C .

We now consider some possible behaviors of the system:

- If only replacement is present ($z > 0, p = 0, \mu = 0$), then we have a default RPS scenario where the population will continually cycle forever without any species gaining a permanent advantage. Whenever an individual beat another, it takes its spot without creating any empty spaces or taking up new empty spaces. This is known as the Lotka–Volterra model.
- If replacement and removal are present ($p > 0, z > 0, \mu = 0$), coexistence becomes unstable and trajectories spiral outwards. The system takes turns between different species dominating for long period of time before being taken over by another in order of their RPS dominance. This is known as the May–Leonard model.
- When we add in mutations ($p > 0, z > 0, \mu > 0$), for small values of μ , we see [Hopf bifurcations](#). Specifically a supercritical Hopf bifurcation arises, since the fixed point of coexistence becomes unstable and a small limit cycle appears around it. As μ is increased, the limit cycle shrinks and collapses into the fixed point, making it stable once again. As a result, large values of μ yield a stable coexistence.

The contents discussed in this section pertain to well-mixed infinite populations. For finite systems, randomness plays a much larger role. Factors such as environmental noise and its resultant fluctuations can cause the system to not follow the cycles as predicted above. As a result, random chance may cause a species to go extinct, which in turns results in one final species emerging as the victor. This final single-species state is called an “absorbing state,” and the time it takes to reach this absorbing state is roughly proportional to the population size, or is $O(N)$. Interestingly enough, it has been shown through computer simulations that in a well-mixed finite system, it is most common for the weakest species, i.e., the species with the smallest rates of dominance-removal p and dominance-replacement z to emerge victorious. This phenomena is referred to as ”survival of the weakest.” One may have issue with guaranteed extinction for finite well-mixed populations. This sets the stage for the next section which talks about how spatial structure, as opposed to systems being well-mixed, restores coexistence.

3.3. Spatial Mobility. In real biological systems one of the most important factors between species is mobility. To account for mobility between the species present in our system, we will assume that individuals can move by swapping positions with their neighbor at some rage γ according to the scheme of

$$\begin{cases} XY \rightarrow YX \\ X\emptyset \rightarrow \emptyset X. \end{cases} \quad (3.3.1)$$

In the beginning of their simulation, Reichenbauch et al. [6] assume a mobility rate m , a rate of competition between species p and a reproduction rate q which are all set to 1. This system

exhibits sensitive dependence on initial conditions. Under the assumption $p = q = 1$ the critical value for mobility can be empirically determined to be $m_c = (4.5 \pm 0.5) \times 10^{-4}$ [8]. If $m < m_c$ then the probability of extinction, where two species disappear after a time proportional to the system size goes to 0, and if $m > m_c$, this probability goes to 1.

Assuming that the lattice spacing $1/N$ vanishes with $N \rightarrow \infty$, the spatial dynamics of the RPS game can be characterized by the following system of differential equations [8]:

$$\begin{cases} \partial_t a(r, t) = D_m \nabla^2 a(r, t) + qa(r, t)\rho_\emptyset - pc(r, t)a(r, t) \\ \partial_t b(r, t) = D_m \nabla^2 b(r, t) + qb(r, t)\rho_\emptyset - pa(r, t)b(r, t) \\ \partial_t c(r, t) = D_m \nabla^2 c(r, t) + qc(r, t)\rho_\emptyset - pb(r, t)c(r, t) \end{cases} \quad (3.3.2)$$

where r denotes the spatial position on the lattice, ρ_\emptyset the density of the empty site and D_m the diffusion coefficient. This set of PDE's is solved numerically to produce the simulation dynamics seen in the bottom row of figure 4.0.1.

3.4. The Complex Ginzburg Landau Equation. We will see how to capture the effect of spatial mobility. We begin by considering the following differential equation

$$\partial_t u = D \partial_x^2 u + F(u) \quad (3.4.1)$$

where D is the diffusion matrix and F is a function of our choice. Equation (3.4.1) is known as the *reaction diffusion equation*. In this system the Hopf bifurcation arises when $\lambda(\mu) = \alpha(\mu) \pm i\omega_0$ where $\alpha(0) = 0$ and $\alpha'(0) \neq 0$ and where $i\omega_0$ represents the slow oscillation for some frequency ω_0 .

We wish to quantify the behavior of oscillations around this bifurcation, we will introduce oscillating terms in both the spatial and temporal directions, $T = \epsilon^2 t$ and $X = x\epsilon$ for some arbitrarily small parameter ϵ which renders these oscillations "slow". We apply the chain rule to these slow oscillation terms and plug them into 3.4.1 to expand our solution as

$$u(x, t) = u_0 + \epsilon[A(X, T)e^{i\omega_0 t}\phi + A^*(X, T)e^{-i\omega_0 t}\phi^*] + O(\epsilon^2) \quad (3.4.2)$$

for complex amplitude $A(X, T)$ and the Hopf eigenvector ϕ . Expanding by powers of ϵ , we find that at order ϵ we obtain a linear eigenvalue problem $\Phi\phi = i\omega_0\phi$ where Φ represents the Jacobian of our function F from 3.4.1 evaluated at u_0 with solution $u_1 = Ae^{i\omega_0 t}\phi$. At order ϵ^2 we begin to introduce second order spatial terms in the $O(\epsilon^2)$ term such as the Laplace operator ($\nabla^2 = \partial_x^2 + \partial_y^2$). This again follows from the application of the chain rule to our slow oscillations and plugging them into equation (3.4.1), namely, $\partial_x = \epsilon\partial_X$ and $\partial_t = \partial_T + \epsilon^2\partial_T$. However, at order ϵ^3 , we obtain resonance terms on the order of $e^{i\omega_0 t}\phi$. Assembling these order ϵ^3 terms yields for the right hand side

$$\partial_T Ae^{i\omega_0 t}\phi + \alpha \nabla^2 Ae^{i\omega_0 t}\phi - \beta|A|^2 Ae^{i\omega_0 t}\phi. \quad (3.4.3)$$

In order for this equation to be solvable, we require that these order ϵ^3 terms be orthogonal to the eigenvector ψ that is adjoint to ϕ the Hopf eigenvector. This implies that

$$\partial_T A = \alpha A + \beta \nabla^2 A - \gamma|A|^2 A. \quad (3.4.4)$$

Now, we can introduce a normalization condition to nondimensionalize 3.4.4 by rescaling $t = \alpha T$, $x = \sqrt{\frac{\alpha}{\beta}}X$ and $A_{new} = \sqrt{\frac{\alpha}{\gamma}}A_{old}$. Equation (3.4.4) then becomes

$$\partial_t A = A + (1 + ic)\nabla^2 A - (1 - ic)|A|^2 A \quad (3.4.5)$$

which is known as the *Complex Ginzburg Landau Equation*. It is this equation that enables us to simulate the spatiotemporal dynamics of the RPS game [8].

4. METHODOLOGY AND TECHNICAL IMPLEMENTATION

Inspired by [8], we carried out GPU-based simulations of the Rock Paper Scissors game. We use python as our programming language, and CuPy for GPU-optimized computations. This is because our update rules are mathematically simple, yet very easily parallelized: each update applied to every cell in our game's grid should be updated synchronously. CuPy lets us exploit the inherent parallelism of the problem. Finally, we run the system for a number of epochs, and download it to visualize locally using PyGame, a game-engine library in python, which we use for its graphics processing capabilities.

We investigate bifurcations that occur when we change parameters in the model. Specifically, we investigate the emergence of cyclic, periodic spirals noted in [8] and [1]. We know from our discussion of attractors and 3.4.5 that there is a bijective correspondence between iterated dynamic system and continuous differential equation formulations of the Rock Paper Scissors game; we use a discrete analogue, and at each step, sample update rules according to 4.1.2 as a random walk process.

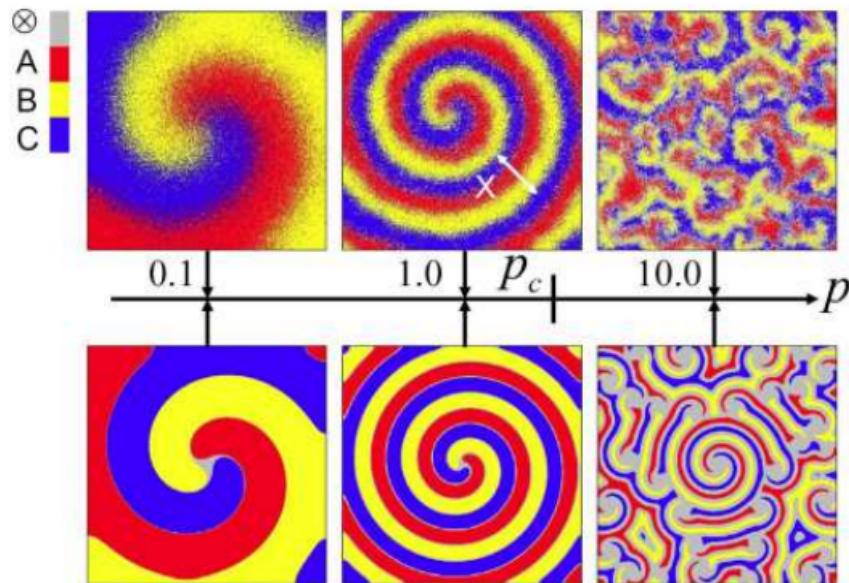
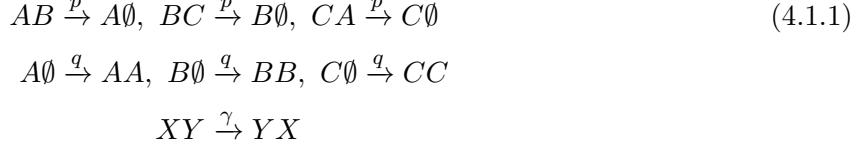


FIGURE 4.0.1. RPS cyclic spatial patterns noted in [1] simulated with random walks and numerically solving PDEs

4.1. Mathematical Dynamics. For our simulation, we adapt section 3.2 as follows:



We use an $n \times m$ grid partition on a 2D lattice. Practically, this is stored as a 2D cupy array of values, where WLOG we assign an integer to each species. Then, we normalize these rates as transition probabilities according to [1, 6] where the probability of a given cell ‘outcompeting’ a neighbor is $p/p + q + \gamma$, the probability of a cell settling a neighbor is $q/p + q + \gamma$ and the probability of a cell switching places with a neighbor is $\gamma/p + q + \gamma$. Neighbors include diagonal-bordering cells, so any interior cell has 8 neighbors, not 4. Thus, we define our transition *probabilities* in terms of these rates as

$$\begin{aligned} \hat{p} &:= \frac{p}{p + q + \gamma}, \\ \hat{q} &:= \frac{q}{p + q + \gamma}, \\ \hat{\gamma} &:= \frac{\gamma}{p + q + \gamma}. \end{aligned} \tag{4.1.2}$$

The crucial thing to note is the existence of γ , a value that enables *mobility*. [8] argue that mobility leads to much more robust mixing, and is a preserving force for species’ co-existence in the game. How do we measure this mobility? In [6], the *mobility/diffusion coefficient* is defined as $2\gamma/N$ for $N := n \cdot m$. Importantly, however, this definition of a mobility coefficient is somewhat moot for our implementation. Consider two settings: for $p, q, \gamma = 10$ and $p, q, \gamma = 1000$, all the actual transition probabilities are equal because we normalize each by the sum of p, q , and γ . However, the Reichenbauch mobility coefficients would be a factor of 100 apart! Thus, we investigate a *normalized* mobility coefficient defined as

$$\nu := \frac{1}{-\ln(\frac{\gamma}{p+q+\gamma})}. \tag{4.1.3}$$

We make this transformation because, as defined in [6], mobility is a *rate* across the whole system, so their mobility coefficient is averaged across each cell to find a per-cell mobility. This form of non-dimensionalization isn’t required for our simulations, because our transition function is a *per-cell probability* so our probability $\hat{\gamma}$ is already independent of system size. Finally, the log term in the denominator transforms mobility to a float with a nicer range.

We investigate different values of this coefficient and how they affect the ‘complexity’ of our system. Quantitatively, ‘complexity’ is measured in *entropy*, defined traditionally as

$$H = - \sum_i p(x_i) \log_2 p(x_i)$$

where p is a random variable over finite support. There are whole branches of philosophy to describe what entropy ‘means’ intuitively, but one way of thinking about it is how unpredictable your

random variable is. In computer science and information theory, this has direct links to how much information it takes on average to represent our random variable.

There are other ways to measure entropy between multiple random variables, such as mutual information and KL divergence. However, we want an entropy measure that captures the ‘intricacy’ of our game. Consider the following two images:

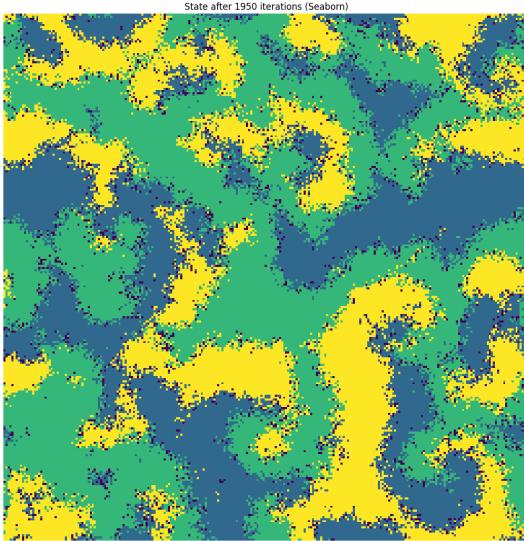


FIGURE 4.1.1. A simpler RPS state.

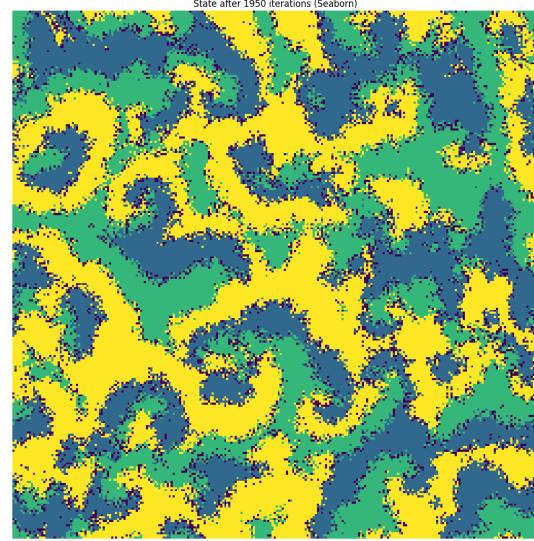


FIGURE 4.1.1. More complex state.

both of these are generated from our RPS implementation. However, one is clearly more ‘complex’ than the other, with a higher intricacy around the borders between populations zones of species. We therefore introduce the following measure of entropy that captures ‘border complexity’. As far as we know, it is a new definition of entropy, and is $O(Nd)$ to calculate, where N is the number cells in the system and d is the number of edge relations for each cell. (Computation can also be understood in terms of a 3x3 sliding window- more complicated measures of entropy could involve convolution of windows of different sizes for example).

Our *Border/Boundary Entropy* C is defined as:

$$C := - \sum_x \sum_{y \in N(x)} p(x) \ln(q(x, y)) \quad (4.1.4)$$

Crucially, $p()$ and $q()$ are probability functions, and different to p, q as competition and settlement rates defined in [4.1.1](#).

We sum over all cells x in our game, and $N(x)$ is a set defining the neighbors of a cell x - in an ‘inner’ cell in our implementation, there are 8 neighbors corresponding to the outer cells in a 3x3 grid. $p(x = r)$ takes the probability of a cell x being type r as the population proportion of the overall system: for a game with 225 total cells, and 21 r type species, $\hat{p}(x = r) = \frac{21}{225}$ for *any* cell

x . Finally, $q(x, y = a, b)$ captures the probability that a cell x of type a has a neighbor y of type b . To understand how $q(x, y)$ behaves, consider the following.

Suppose $x = r$, meaning x is rock. We assume that $q(x, r) = 1$, meaning we expect a rock to be neighbors with another rock. This zeros out in our logarithm, contributing nothing to the border entropy of our system (as this is not a border between species). Also, $q(x, e) = 1 - \hat{q}$ for e an empty cell and \hat{q} our settlement probability defined in 4.1.2. Finally, for cross border relations, $q(x, s) = q(x, p) = 1 - \hat{p}$. Crucially, $q()$ is symmetric both across different species-relations and across borders, meaning that $q(x,y) \equiv q(y,x)$.

For games with different competition rates between species, our function $q()$ would have to be defined differently, taking into account different competition probabilities. Thus, border entropy scales with different competition and settlement probabilities, and the 'likelihood' a cell has a neighbor that either it dominates or dominates it, scales with this probability of competition.

4.2. Algorithmic Logic. Our simulation consists of two main functions. Both make use of CuPy's parallelism and broadcasting functionality. Thus, most variables are stored as CuPy tensors, and most functions are done through masks.

The first function, seeding, takes in an empty grid and a density, and outputs a grid with randomly distributed species according to the density given. The density is passed to a Bernoulli sample over each grid cell, and cells are correspondingly triggered.

Algorithm 2 seeding

Require: grid ($n \times m$ integer lattice), density (float between 0 and 1)

Ensure: gridSize

```

1: for cell in grid do
2:   value  $\sim U := (U_1(1, 2, 3), U_2(1, 2, 3), \dots, U_{|grid|}(1, 2, 3))$ 
3:   toggle  $\sim B := (Bern_1(\text{density}), Bern_2(\text{density}), \dots, Bern_{|grid|}(\text{density}))$ 
4:   if valuesi is togglei then
5:     gridi  $\leftarrow$  togglei                                 $\triangleright$  mapping the ith cell to the ith entry of toggle
6:   end if
7: end for
```

Our next algorithm updates our game state. The internal logic has been simplified for ease of reading, but the core of the algorithm is the fact that the three probability values, \hat{p} , \hat{q} and $\hat{\gamma}$ all sum to one. Thus, they can be seen as a tripartition of the interval $[0, 1]$. This lets us sample this interval to make cell-wise decisions. Again, this is implemented using masks through CuPy broadcasting functionality.

Algorithm 3 update

Require: grid ($n \times m$ integer lattice), [settle,competitor,mobility] (tuple of float values between 0,1 s.t. their sum is 1)

- 1: **for** every cell in grid **do**
- 2: cast $\leftarrow U(0,1)$
- 3: decision $\leftarrow \begin{cases} s & \text{if cast} < \text{settle} \\ c & \text{if settle} \leq \text{cast} < \text{compete} \\ m & \text{if compete} \leq \text{cast} < \text{mobility} \end{cases}$
- 4: **if** decision = s and cell is not empty **then**
- 5: empty random neighboring cell \leftarrow cell
- 6: **end if**
- 7: **if** decision = c and cell is nonempty with outcompeted neighbor **then**
- 8: out competed cell \leftarrow empty
- 9: **end if**
- 10: **if** decision = m **then**
- 11: neighbor \leftarrow randomly chosen neighbor
- 12: switch(neighbor,cell)
- 13: **end if**
- 14: **end for**

In practice, `update` is parallelized, so all cell updates happen synchronously. This prevents any one cell from being updated before another, preventing cell update hierarchy leading to invalid results. For code implementation, visit [our github](#).

4.3. Data Collection and Analysis. For our system size, we use grids of size 256x256 and then 512x512. Thus, $\sqrt{N} = 256, 512$ for different simulations. We use powers of two to mesh better with SIMD architecture, so it is easier for threads to stay synchronized. We simulate a parallel implementation of RPS by initializing the grid according to some density, and then running it for a number of iterations. For measuring entropy, we first sample a number of mobility coefficients (4.1.3) over a range, and then calculate border entropy as defined in 4.1.4 in parallel over all cells, then summing all cells' entropic contributions. We also investigate population counts over time. All methods can be found on the [github](#)'s colab notebook.

4.4. Results. Our first noteworthy qualitative results is that species extinction never occurred despite a large range of parameters being tested. This would suggest that if species extinction occurs at all, it a globally unstable attractor. Regardless of parameter values, population seemed to stabilize with two main periodic patterns, one of high amplitude, and a smaller period of lower amplitude population cyclicity.

Second, is that convergence to single-arm spiral phenomena, like in figure 4.0.1, never occurred. We suspect this is because our model calculates transition probabilities with *normalized* probabilities. That implies transition probabilities are always large with respect to one another, so parameter ranges where single armed spirals occur are never reached.

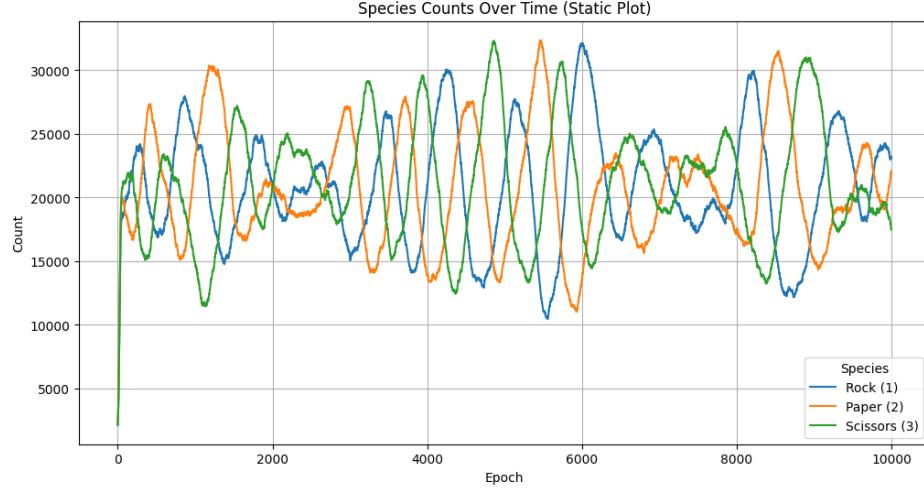


FIGURE 4.4.1. Population over time.

For our main results, we investigate how border entropy is influenced by different mobility coefficients, as defined in equations 4.1.4 and 4.1.3. We search through 81 different parameter values of γ after fixing p and q , as defined as ‘rates’ in equation (4.1.1). This means that $\hat{p} = \hat{q}$ and we iterate over slices of ν . For each parameter value γ_i , we run five different game instances, and wait 1000 iterations for transient periods to stabilize. We then average these 5 different entropy values, and add it to a plot. The variance among entropies of games with the same parameters varied very little.

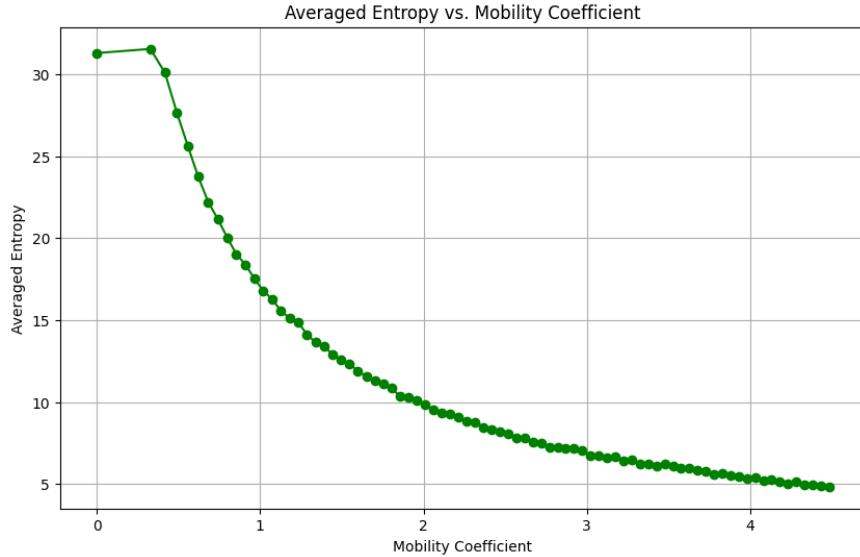
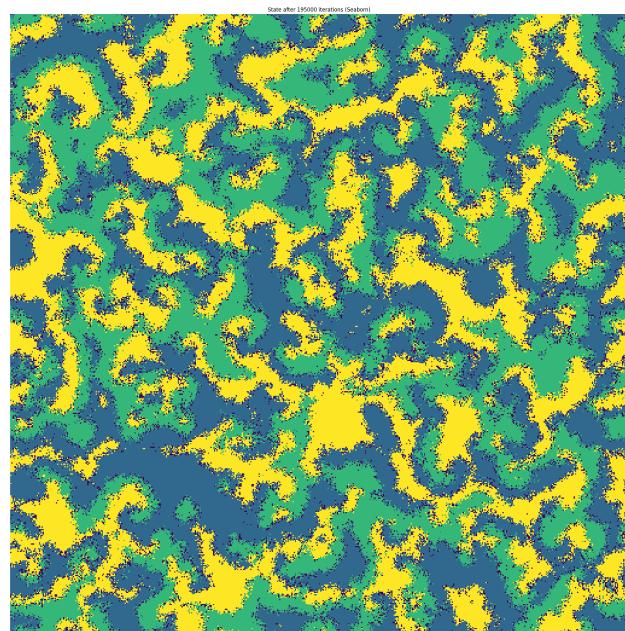
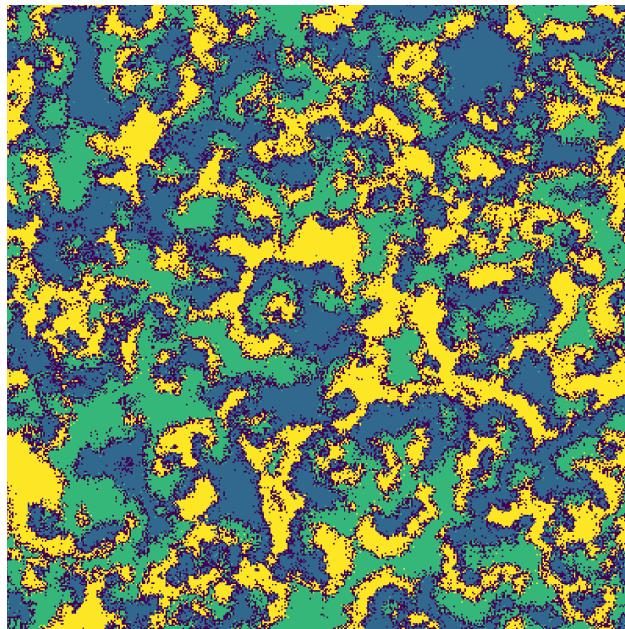


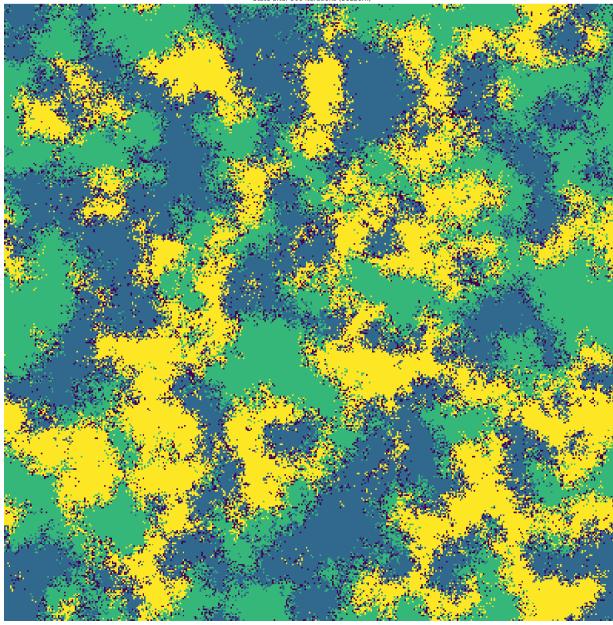
FIGURE 4.4.2. Border entropy against normalized mobility.

We see that boundary entropy has a strong negative correlation, shaped like a negative exponential with mobility. This makes sense - the \ln component of C would shrink as \hat{p} and \hat{q} shrink.

Thus, normalizing transition variables makes the game less entropic as mobility increases.

Mathematically, this can be understood with reference to the random variable defined in algorithm 3 4.2: the entropy of our tripartition random variable of $[0,1)$ decreases logarithmically when we increase $\hat{\gamma}$. Therefore, in an RPS system where choices of action are uniform and conserved according to their rates, the entropy of the overall system scales with the entropy of this action choice. The entropy of our choice directly affects the entropy of the game system. Finally, we produced some pretty art.





5. CONCLUSION

In this report, we used tools from nonlinear dynamical systems, chaos theory, and evolutionary game theory to study cyclic dominance and rock-paper-scissors systems. We laid out the mathematical background needed to analyze these systems. Some topics include fixed points, phase portraits, bifurcations, linearization, Jacobian matrix, limit cycles, Hopf bifurcations, strange attractors, and more.

We then reviewed a paper by Szolnoki et al. [8] on cyclic dominance. We studied both well-mixed systems and systems with spatial mobility. We laid out the mathematical framework of RPS games, defining key constants such as dominance-replacement, dominance-removal, reproduction, and mutation. We covered the intuition behind the differential equations of the system, and covered possible behaviors of the system when we change these said parameters. Spatial mobility changes the picture greatly, and topics such as the system's reaction-diffusion equations were reviewed. We reviewed how when the parameter nears its Hopf bifurcation point, the spirals and oscillations can be modeled by the Complex Ginzburg Landau Equation.

Finally, we coded a parallel framework to simulate RPS games in python and CuPy. We derive a new form of entropy that captures the intuitive ‘complexity’ of a game state, and run simulations that relate this form of entropy to mobility, demonstrating a relation between the entropy of the game and the entropy of the underlying choice random variable.

REFERENCES

- [1] Luo-Luo Jiang et al. “Effects of competition on pattern formation in the rock-paper-scissors game”. In: *Physical Review E* 84.2 (Aug. 2011). ISSN: 1550-2376. DOI: [10.1103/physreve.84.021912](https://doi.org/10.1103/physreve.84.021912). URL: <http://dx.doi.org/10.1103/PhysRevE.84.021912>.
- [2] S. Lynch. *Dynamical Systems with Applications using MATLAB®*. Birkhäuser Boston, 2004. ISBN: 9780817643218. URL: <https://books.google.com/books?id=EIgTh2IBXCgC>.

- [3] Alex Mac. *Why does the Mandelbrot Set only bifurcate on real values?* Mathematics Stack Exchange question asked 25 Sep 2021. 2021. URL: <https://math.stackexchange.com/questions/4260245/why-does-the-mandelbrot-set-only-bifurcate-on-real-values> (visited on 11/24/2025).
- [4] Stanford University Department of Mathematics. *Differential Equations with Linear Algebra, Fourier Methods, and Modern Applications*. Course text for Math 53, last modified September 22, 2025. Stanford, CA: Stanford University, 2025.
- [5] Malihe Molaie et al. “Simple Chaotic Flows with One Stable Equilibrium”. In: *International Journal of Bifurcation and Chaos* 23.11 (Nov. 2013), p. 1350188. DOI: [10.1142/S0218127413501885](https://doi.org/10.1142/S0218127413501885).
- [6] Tobias Reichenbach, Mauro Mobilia, and Erwin Frey. “Mobility promotes and jeopardizes biodiversity in rock–paper–scissors games”. In: *Nature* 448.7157 (Aug. 2007), pp. 1046–1049. ISSN: 1476-4687. DOI: [10.1038/nature06095](https://doi.org/10.1038/nature06095). URL: <http://dx.doi.org/10.1038/nature06095>.
- [7] Steven H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Westview Press, 2000.
- [8] Attila Szolnoki et al. “Cyclic dominance in evolutionary games: a review”. In: *Journal of the Royal Society Interface* 11.100 (2014), p. 20140735. DOI: [10.1098/rsif.2014.0735](https://doi.org/10.1098/rsif.2014.0735). URL: <http://dx.doi.org/10.1098/rsif.2014.0735>.