



# Ecosystem Simulation Study

Joan Pascual Ribes & Eki González Flamarique



June 12, 2024

## **0.1 Abstract**

The Ecosystem Simulation Study (ESS) project presents a Python-based simulation framework designed to model complex interactions within an ecosystem. Through the simulation, entities known as "blobs" interact within a grid environment, exhibiting behaviors such as movement, predation, reproduction, and adaptation. This project is thought to be long-term work, so this paper provides an overview of the project's objectives, methodology, key components, and some of the already-attained achievements, offering insights into emergent behaviors and ecological dynamics and the promising project's progress.

## **0.2 Personal Interest**

The Ecosystem Simulation Study project holds personal interest for us due to its interdisciplinary nature and potential applications in both scientific research and education. As people who are passionate about ecology and computational modeling, we are excited about the opportunity to explore complex ecological processes in a virtual environment. Furthermore, we are strongly interested in programming, thermodynamics, chaos theory, game theory, mathematics, and statistics, which have been useful tools during the development. We believe that the insights gained from this project can contribute to our understanding of ecosystem dynamics and inform strategies for environmental conservation and management.

# Contents

0.1 Abstract . . . . .	1
0.2 Personal Interest . . . . .	1
<b>1 Introduction</b>	<b>3</b>
1.1 Description of the tool . . . . .	3
1.2 Objectives and research methodology . . . . .	5
<b>2 Dynamics of the two-species system</b>	<b>5</b>
2.1 Lotka-Volterra equations . . . . .	5
2.2 Sparse Identification of Nonlinear Dynamical Systems . . . . .	6
2.3 First results and comments . . . . .	7
2.4 Unpredictability of the system according to initial conditions . . . . .	10
2.5 The buffering effect . . . . .	13
2.5.1 Study of dynamics . . . . .	14
2.5.2 Comparison with Lotka-Volterra . . . . .	17
2.5.3 Cause hypothesis . . . . .	18
2.6 Periodicity & frequency analysis . . . . .	19
2.6.1 Fourier analysis . . . . .	19
2.6.2 Frequency study of the data . . . . .	19
2.7 Evaluation of model consistency and discrepancies with the LV model . . . . .	21
<b>3 Dynamics of one-species systems</b>	<b>23</b>
3.1 Logistic growth and decay . . . . .	23
<b>4 Genetic adaptability and species emergence</b>	<b>25</b>
4.1 Density-Based Spatial Clustering of Applications with Noise . . . . .	26
4.2 Emergence of new species from Phyto . . . . .	26
4.3 Genetic adaptability and ecosystem sanitation . . . . .	27
4.3.1 The importance of genetic variability . . . . .	30
<b>5 Comments, difficulties, and observations</b>	<b>30</b>
5.1 The complex predation balance . . . . .	30
5.2 Dependence on coordinates of SINDy . . . . .	31
5.3 Comments about the code . . . . .	32
<b>6 Future objectives</b>	<b>33</b>
6.1 Trophic web . . . . .	33
6.2 Soil . . . . .	34
6.3 Going deeper with the dynamic study . . . . .	35
<b>7 Conclusion</b>	<b>35</b>
<b>8 Bibliography</b>	<b>36</b>
8.1 Acknowledgements . . . . .	36

# 1 Introduction

## 1.1 Description of the tool

We developed our own tool in Python that allows simulating an ecosystem formed by entities called “blobs” that have different attributes that govern their behavior and interactions. These properties are systematically defined to provide a global understanding of their characteristics and dynamics.

Blobs lay in a discrete closed world where borders are identified, they navigate through it via theirs spatial orientation which is delineated by its *position*, denoted by horizontal (*x*) and vertical (*y*) coordinates. These coordinates serve as foundational parameters for navigation and spatial relationships within the simulated environment.

This world has *Energy* which serves as a vital resource for blobs, dictating their vitality and capacity for reproduction. Blobs lose energy according to their metabolism (which depends on the development of physical features) and when they move.

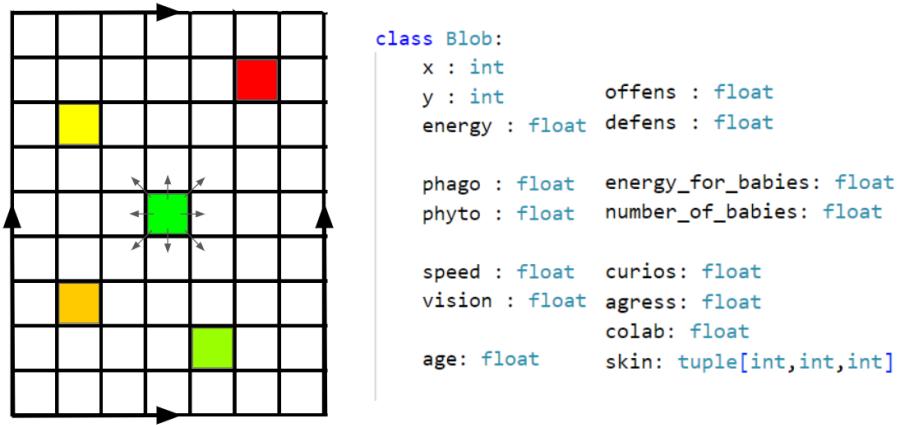


Figure 1: Schematic representation of blobs in their environment, together with the attributes of the Blob class.

The features of a blob are :

- Feeding preferences elucidate the dietary inclinations of blobs, encompassing two dimensions: *phago* and *phyto*. *Phago* quantifies the efficacy and capacity of prey consumption in energizing the blob, while *phyto* measures its ability to derive sustenance from the environment (autotrophic nutrition). These preferences guide the foraging behavior and resource acquisition strategies of each blob. For visual representation, we would indicate a blob’s *phyto* preferences by coloring it with green scale and *phago* with red scale. See Figure 1.
- Movement parameters encompass *speed* and *vision*, essential components of locomotion and perception within the simulated ecosystem. Speed denotes the velocity at which a blob tra-

verses its environment, while vision defines its visual range and acuity. These properties contribute to the efficiency of movement and the detection of environmental stimuli.

- *Age* represents the temporal dimension of blob existence, indicating the elapsed duration (number of iterations) since inception. Blobs possess a predetermined maximum age, beyond which its lifecycle concludes. Age serves as a critical factor in population and evolutionary processes.
- Combat abilities assess the offensive and defensive capabilities of blobs in interactions with other entities. *Offens* quantifies the efficacy of capturing and consuming prey, while *defens* measures the resilience of the blob in withstanding attacks from predators. These attributes influence the survival and reproductive success of individual blobs.
- Reproductive parameters govern the propagation of blobs within the ecosystem. *Energy\_for\_babies* determines the allocation of resources to offspring, while *number\_of\_babies* denotes the reproductive output of each entity. These parameters contributes to the variety of reproductive strategies.
- Behavioral traits encompass intrinsic tendencies and responses exhibited by blobs. *Curios* reflects the propensity for exploratory behavior, while *agress* signifies the inclination towards pursuit or confrontation, and *colab* indicates the propensity for cooperative interactions among individuals with similar genetic characteristics.
- Appearance attributes define the visual characteristics of each blob, including skin color encoded as RGB values. These features contribute to individual recognition and species identification.

The functioning of the ecosystem is governed by hyper-parameters, global variables that affect transversely the functioning of the simulation. These are:

- The internal energy consumption is calculated by averaging the physical characteristics and scaled by the metabolic constant *metabolism* which regulates the overall consumption and allows us to adjust it.
- The *energy\_to\_reproduce* parameter is used to decide when a blob spawns. When the energy of a blob exceeds this threshold, it reproduces according to its reproductive strategy.
- *Phytogain* regulates the energy input to the system by iteration (emulating solar irradiation). A blob will get at most this energy, and will get less if it has low phyto (adaptation) and/or if it is surrounded by other blobs (resource competition).
- The generic variability is encoded by *gen\_var*. It represents the noise introduced in the genetic code of the offspring with respect to that of its parent. This noise is no more than a normal distribution with this parameter as standar deviation.

Also, there is a auxiliary class called Grid which represents the environment in which blobs reside. It manages the spatial arrangement of blobs and provides methods for querying neighboring blobs and updating their positions.

The simulation is executed using Pygame [6], a Python library for creating interactive applications, that allows us to see in real time what the blobs are doing via its graphic interface as well as

update hyper-parameters at ease, which can be used to depicting natural disasters.

Finally, when the simulation ends the information on the evolution of the parameters during the simulation and other post-analysis data is plotted [7].

## 1.2 Objectives and research methodology

This paper focuses on examining the validity and robustness of the Python-based simulation framework developed so far. For this purpose, population dynamics phenomena with extensive literature, such as predator-prey relationships or just one-species system [5], will be simulated and the extracted data compared with theoretical behavior through the use of a variety of data analysis tools.

During the development process, several ideas and tools have been tested, of which just a few came up to be the most suitable ones. To overcome this challenge, we have applied Agile methodologies to organize ourselves to ensure that the work is completed successfully [10].

We would also like to explore different techniques used in this field, to learn about them and become familiar with their use. Many of them are analytical tools that cut across different areas of expertise, and we believe there is value in getting to know them.

# 2 Dynamics of the two-species system

We will study the dynamics that emerge from an ecosystem composed of two species in the simulation, since it is a topic with extensive literature on which we can rely and of relative simplicity.

In this section, in order to reduce the complexity of the system, we tune ESS to create a submodel that eliminates properties and parameters such as *offens*, *defens*, *agress*, *age*, and *curios*, while fixing *phago* and *phyto* of the two species involved. The hyper-parameters are kept constant over the course of data collection, and in particular, the genetic variability is disabled.

## 2.1 Lotka-Volterra equations

The predator-prey model independently developed by Alfred J. Lotka in 1925 and Vito Volterra in 1926 consists of a system of nonlinear first-order ordinal differential equations that describes the dynamics of a wide range of biological systems where two species (either chemical substances or groups of animals)[2] interact in a cyclic fashion.

The basic idea behind the Lotka-Volterra model (LV) is to understand how the populations of two interacting species change over time due to predation and competition for resources.

$$\frac{dx}{dt} = \alpha x - \beta xy \quad \frac{dy}{dt} = -\gamma y + \delta xy$$

1.  $\alpha$  is the natural growth rate of the prey population.
2.  $\beta$  is the predation rate coefficient, which describes how many prey are killed by a predator.
3.  $\gamma$  is the natural death rate of the predator population.
4.  $\delta$  is the efficiency of turning prey into new predators.

The  $x$  and  $y$  model the natural processes of each species and the terms  $xy$  relate the interaction between species.

This model is simple and certainly ideal, but provides a simple explanation of complex dynamics. For this reason, we will compare the results obtained with this model.

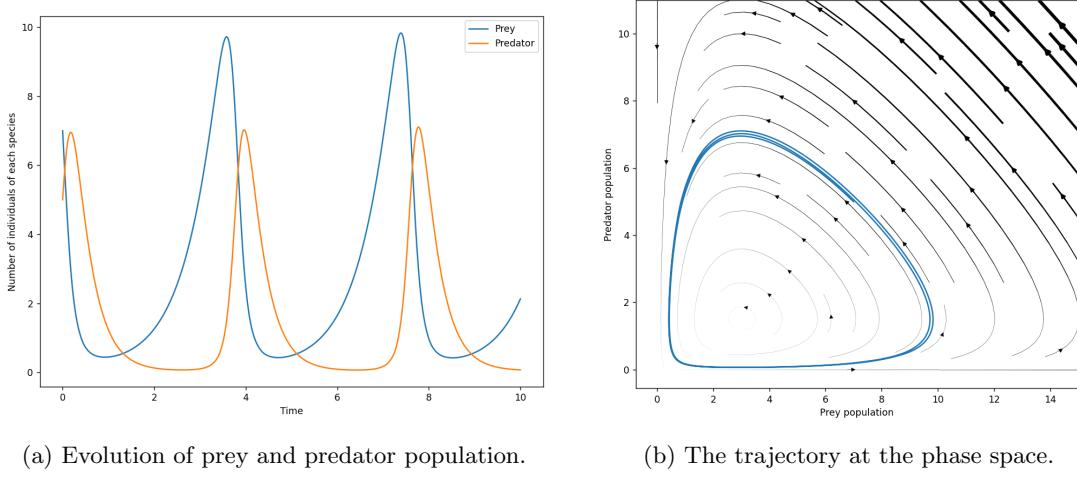


Figure 2: Visualization of a particular trajectory in LV system via numerical integration ( $x_0 = 7$ ,  $y_0 = 5$ ,  $\alpha = 1.5$ ,  $\beta = 1$ ,  $\gamma = 3$ ,  $\delta = 1$ )

## 2.2 Sparse Identification of Nonlinear Dynamical Systems

For the analysis of the dynamics of two species, a algorithm capable of identifying the system's dynamics is required.

Sparse Identification of Nonlinear Dynamical Systems (SINDy) [3] has emerged as a powerful data-driven method for uncovering the underlying equations governing the behavior of complex systems directly from observational data.

This approach, rooted in the fields of dynamical systems and nonlinear dynamics, leverages the sparsity of the system's dynamics to identify the most relevant terms in the governing equations. The core principle of SINDy involves expressing the dynamics of the system as a sparse

combination of basis functions content in what is called a library, enabling efficient extraction of the system's nonlinear dynamics. By integrating techniques from compressed sensing and machine learning, SINDy can effectively identify the governing equations even in the presence of noise and high-dimensional data.

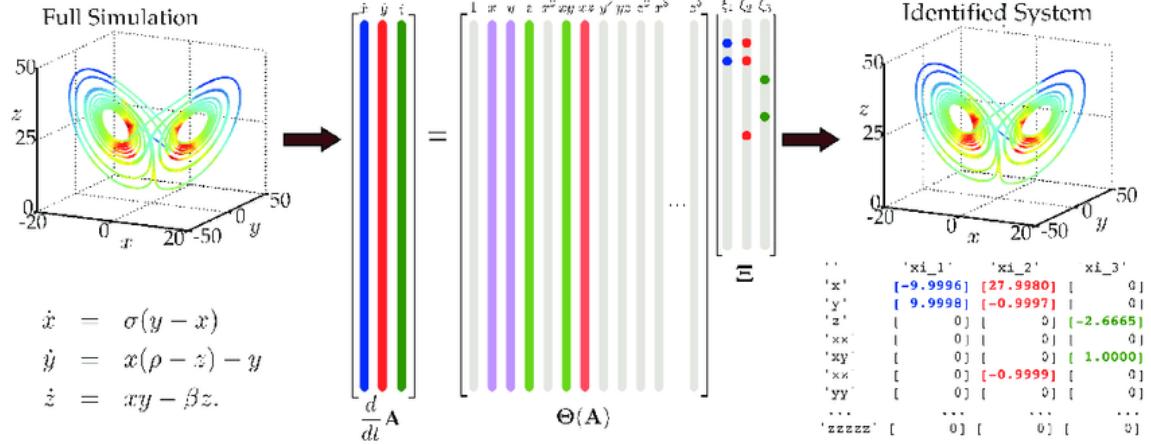


Figure 3: SINDy Model

In this paper, we will use PySINDy[8], an open-source Python library designed to facilitate the implementation and application of the SINDy algorithm. The library provides utilities for model validation, visualization, and comparison, enabling us to assess the quality and reliability of the identified dynamical models.

### 2.3 First results and comments

We prepared a submodel in codes ready to execute, and after costly data compilation work at FIB (Facultat d'Informàtica de Barcelona), we obtained a database of around 8 simulations per possible pair  $(x_0, y_0)$ , where  $x_0, y_0$  are the initial number of autotrophs and heterotrophs, respectively. The database spans from  $(0, 0)$  to  $(500, 500)$  with increments of 50 (a total of 121 possible pairs of initial conditions).

We emphasize that it is needed to perform different simulations for the same pair since the evolution of the system will also depend on the initial positions of the individuals or the random processes that influence their resource-obtaining or decision-making.

As a first observation, the data resembles the LV cyclic pattern quite clearly in the vast majority of the data collected.

Here are some figures of specific moments of the simulation shown below, together with the plot of populations versus time. This is a simulation outside the dataset, with a smaller grid and an initial population of 500 *phytos* and 150 *phagos*.

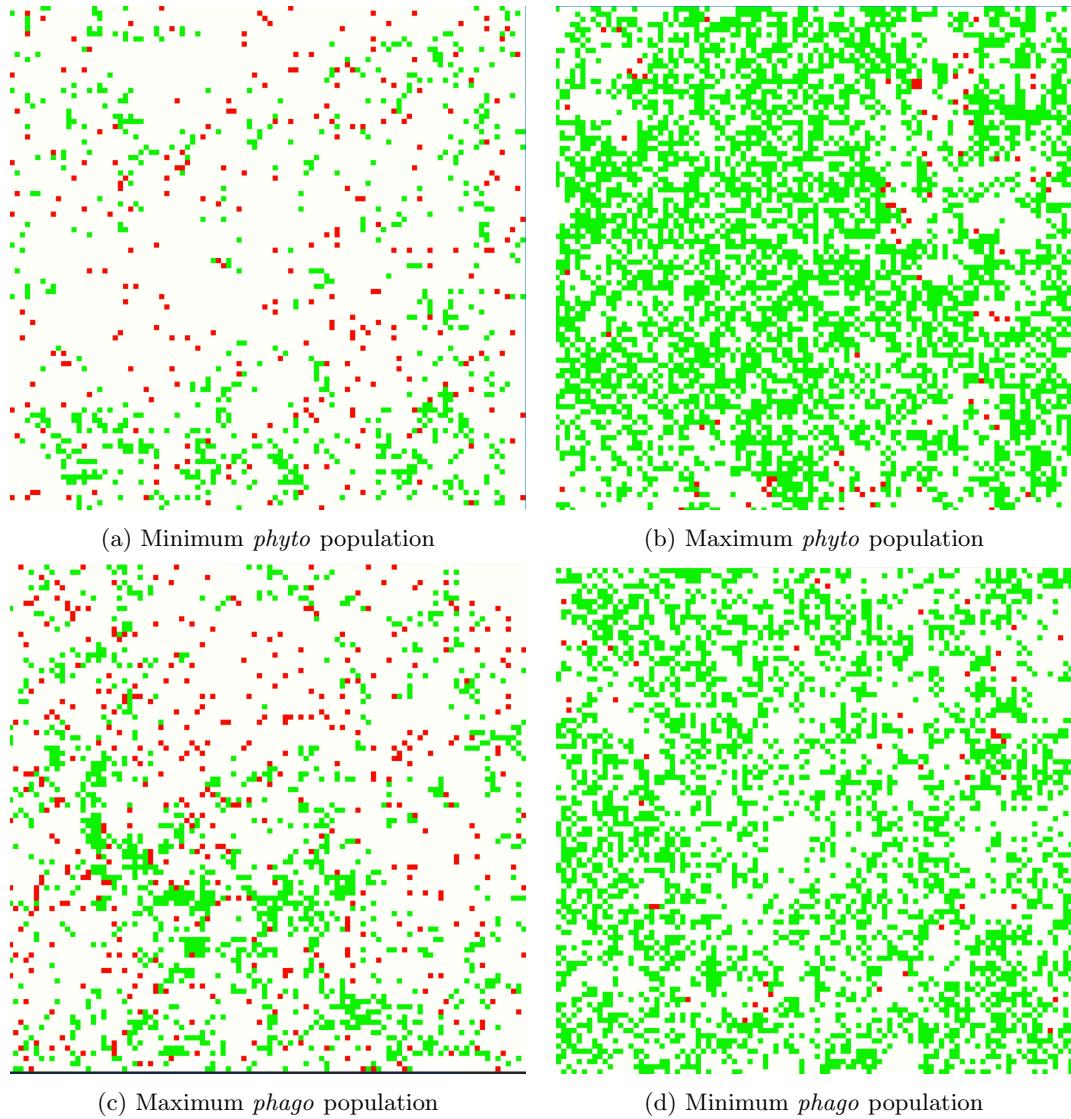


Figure 4: Critical states of the ecosystem

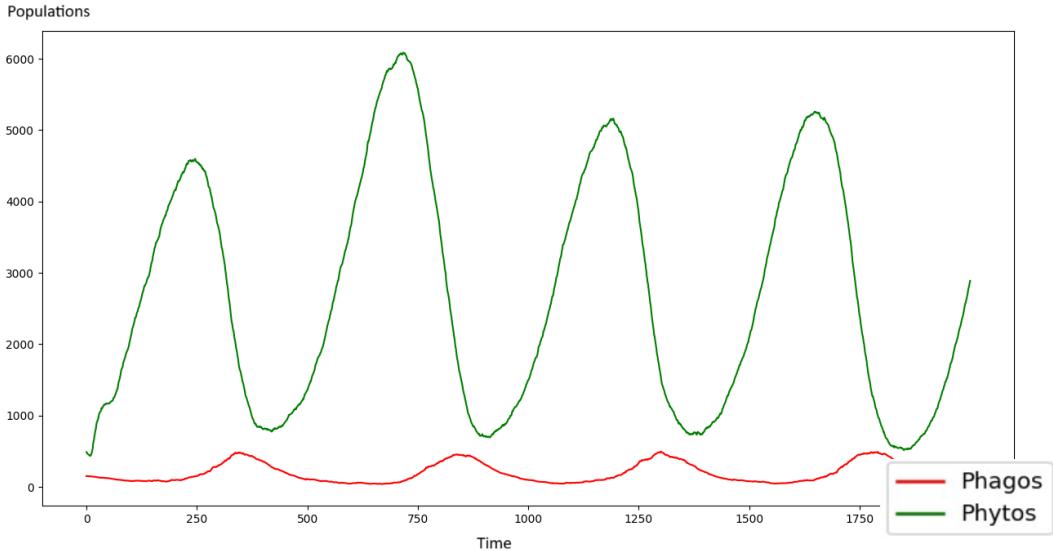


Figure 5: Temporal evolution of populations

As can be seen in Figure 5, the curves agree with high fidelity with a classical LV. The description of the four moments represented in Figure 4 follows.

- In this image, we observe a minimum in the phytos population, which is set at 808 individuals, being reached at iteration 424. At this time, there are 263 predators. The excess of predators in the previous stage causes it to decrease to such low values.
- In this snapshot, we can notice that a maximum is reached in the phytos population due to the low density of predators in the preceding stage, leaving the way free for exponential growth (we will discuss this later). It is reached at iteration 722 with 6068 *phyto* and 90 *phago*.
- In the figure, we contemplate a maximum in the *phago* population reached at iteration 845, with 1609 herbivores and 446 carnivores. It is clearly seen that this maximum is due to the high concentration of prey in the previous stage.
- Here we see a population minimum *phago* reached at iteration 1568 with 3979 autotrophic creatures and only 48 predators. This happens because of the low number of herbivores in the preceding section.

We clearly note that the two populations are not coordinated, although they do have the same frequency (the fact that the relationships involve derivatives stands out here; the differential nature of the model justifies the observations). The stages of growth of the *phytos* population overlap with those of the fall of the *phagos*, influencing each other.

It is also easy to see that the exact same maxims are not always reached. This is due to the irregular behavior that we justified above with respect to the dependence of the positions of the populations and how, in different zones, the carnivore and herbivore population rises are synchronized or not. To exemplify this, we can observe in Figure 6 how, with the same initial number of autotrophs and heterotrophs, the initial pseudo-random spatial arrangement influences the numerical value of maxima and minima.

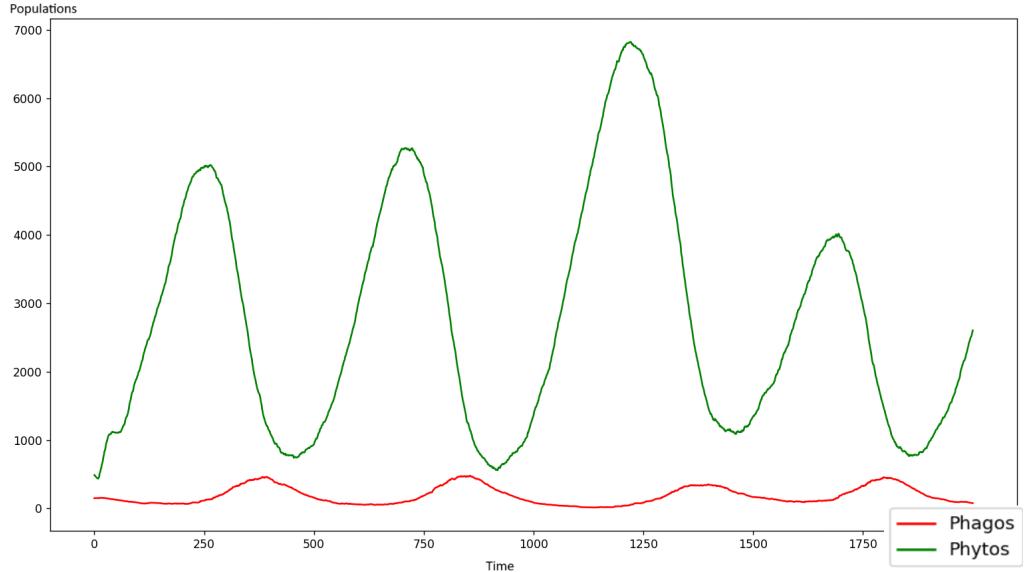


Figure 6: Temporal evolution of populations (II)

These complex behaviors are discussed below.

## 2.4 Unpredictability of the system according to initial conditions

As an object of study in this section, we consider the variance in the data for each pair of initial population pairs, i.e., for the same initial conditions, the curves differ.

This unpredictability phenomenon arises due to the uneven distribution of blobs across the environment when they propagate through space. Initially, blobs are initialized at uniformly random positions, but as time goes on in some regions of the grid, a group of heterotrophs starts to expand as they feed on the autotrophs they encounter, and in those regions without many heterotrophs, autotrophs will thrive, creating a wave-like propagation effect. In this sense, given a certain location, a high concentration of autotrophs will be followed by a high concentration of heterotrophs as they chase autotrophs, and finally a low blob concentration. This region will be slowly inhabited again, and the cycle will repeat. As these heterotrophs waves propagate, they may interfere constructively or destructively, causing small fluctuations in the already oscillating population curves.

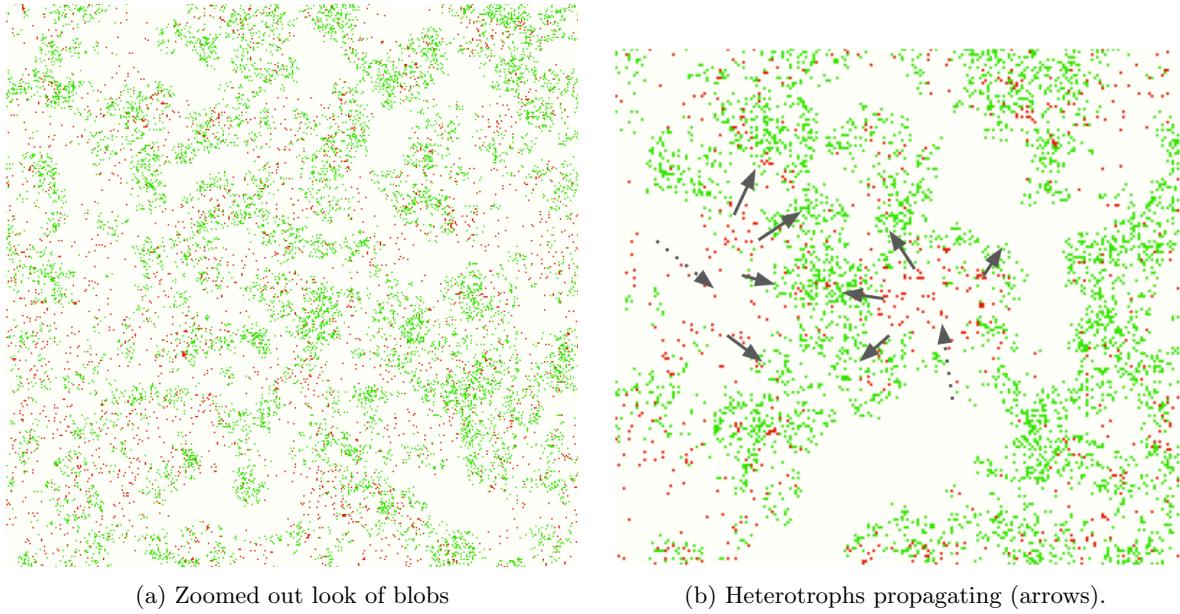


Figure 7: Snapshots of two moments of the simulation

As we can see in Figure 7, at a given instance, the distribution of blobs is quite uneven and dynamic, as the distribution changes by the explained mechanism. In addition, it should also be clear that in the methods (in the code) that regulate the behavior of the blobs, there are functions that have a random output, which contributes to the unpredictability of the dynamics.

Leaving aside this qualitative argumentation, we would like to measure for each initial population (which has a large impact on the early stages) how far the population curves are from each other. To achieve this, according to the data collected for the initial population set, we will perform Principal Component Analysis (PCA) [9] and extract the explained variance ratio of the first component, which will give us a measure of how similar each simulation run is for the same initial conditions.

Plotting the results for the data set, we get the following:

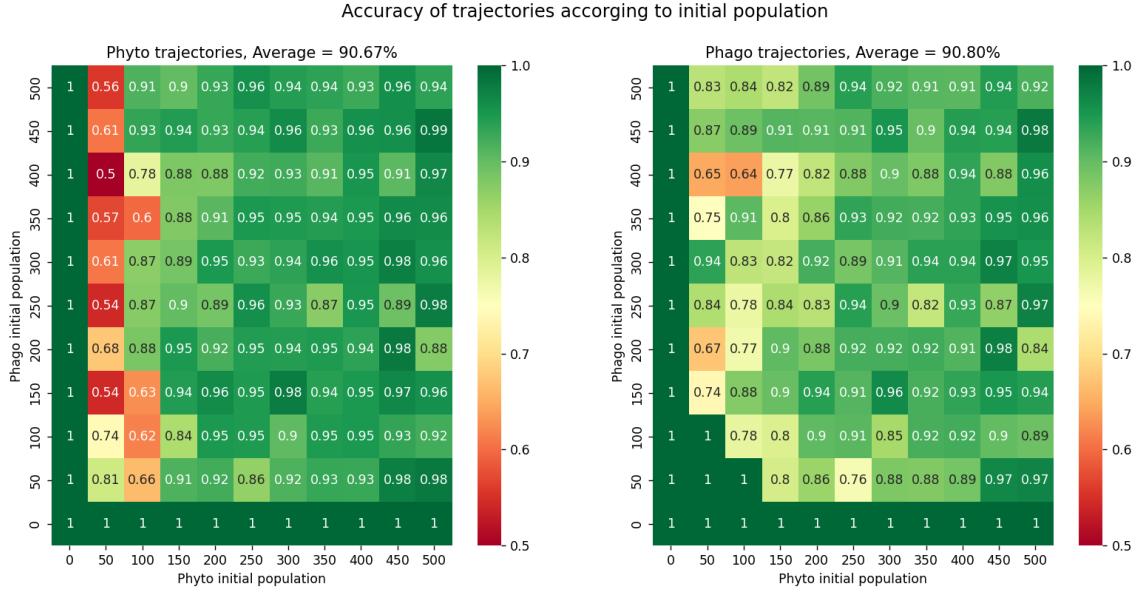


Figure 8: Percentage of similarity of trajectories according to initial populations segregated by phyto and phago.

As is apparent, the simulated trajectories tend to be quite similar, with a high average of 90% for both phyto and phago datasets. In extreme cases where a species begins extinct, simulitude soars. This may be because when autotrophs become extinct, heterotrophs will sooner or later become extinct, giving a constant series of zeros. Analogously, when heterotrophs start to become extinct, autotrophs grow unparalleled until they reach maximum capacity and the population stabilizes at those values.

Another observation that we can extract is that for initial populations of few autotrophs and heterotrophic arbitarius, the simulitude has its lowest values, up to 50%. When observing the numerical series, it can be seen that the population of producers starts so low that in some simulations it ends up becoming extinct or, in others, thrives enough for the system to enter the dynamics of two species, thus distinguishing two quite different scenarios and therefore having a low similarity.

Overall, we have seen that the data obtained is quite consistent and varied, which is recommended for training a model by machine learning, as we will do in the next section, as this avoids overfitting.

Having clarified this, let us study other observed phenomena.

## 2.5 The buffering effect

Another observed phenomenon that we have encountered during practically the entire process is directly related to the dynamics of the system: the gradual decline of the total population peaks.

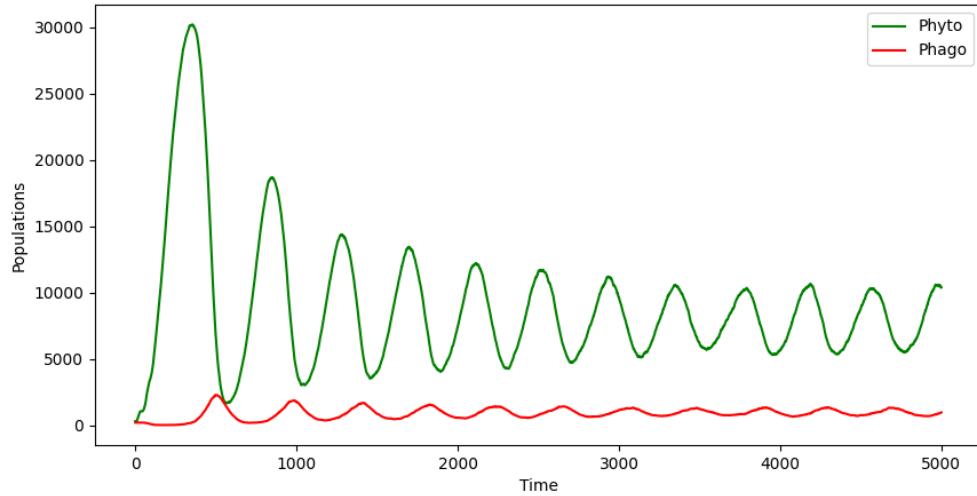


Figure 9: Phenomenon of population buffering

The Lotka-Volterra model consists of closed oscillating trajectories, none of which have a decay in their amplitude. So the cause of this phenomenon must come from another place.

The reader may notice that this pronounced behavior does not occur in the figures of previous sections. This is because the version of the code used is not the same. In this one, a much wider grid is taken, which definitely affects the occurrence of this effect.

### 2.5.1 Study of dynamics

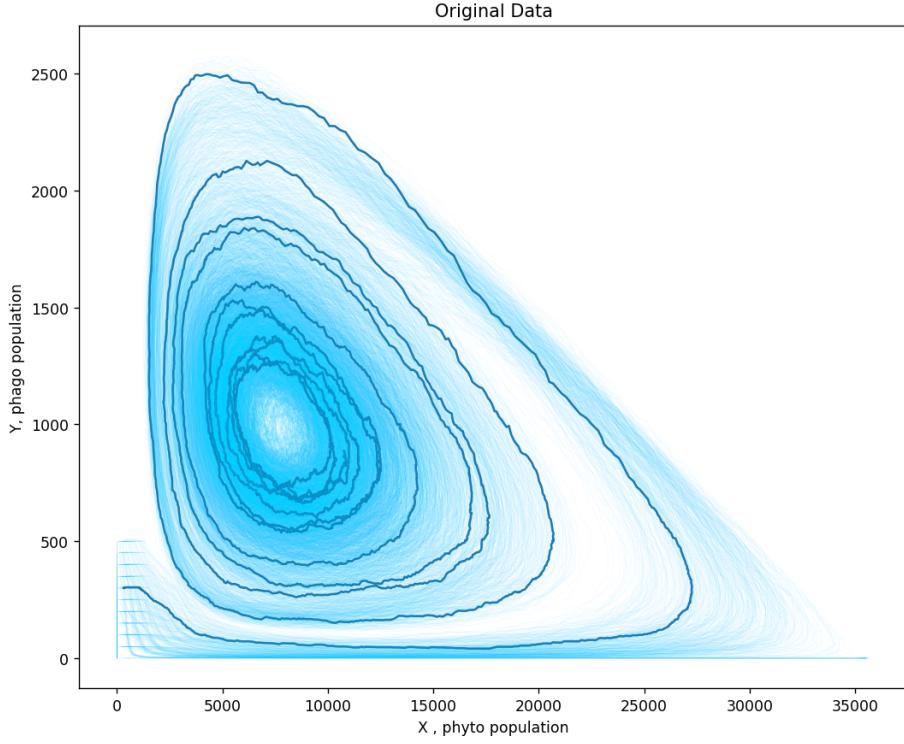


Figure 10: The whole data set is plotted, and a highlighted trajectory.

As evidenced by the plot above, the trajectories really seem to have orbits in the phase space, and those orbits tend to shrink, which can explain the buffering effect.

Let's take a closer look at the dynamics.

First and foremost, let's recall that a spiral down motion in a dynamical system corresponds to a stable point (also known as sink), which can be identified by calculating the eigenvalues of the Jacobian matrix evaluated at that point, as the eigenvalues of a spiral sink must be complex conjugates with a negative real part.

Therefore, by studying the system by means of SINDy, we will obtain the equations of the dynamical system, and after that, we could study its stability.

We will train the model with the whole data set. As a library choice, since it must have a behavior similar to the LV a model, search by polynomial up to quadratic terms will be enabled. It has also been necessary to change the coordinates by normalizing the data in order to facilitate the correct functioning of SINDy (we will develop this problematic in subsection 5.2).

The results are impressive. We can see how the cleaned data match the vector field given by the equations.

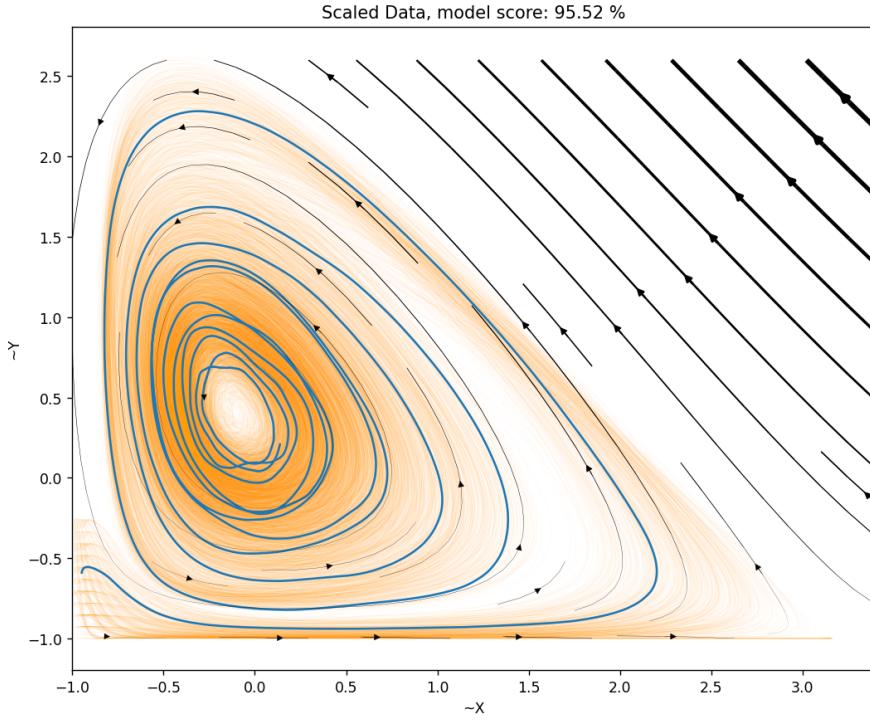


Figure 11: Cleaned and normalized data plotted with same trajectory highlighted, with the vector field.

Once the equations are extracted, a change of variables is realized to return to the original coordinates, and the final equations are obtained. This is the identified nonlinear dynamic system:

$$\begin{cases} \dot{x} = -4.665 \times 10^{-5} x^2 - 2.938 \times 10^{-3} y^2 - 1.444 \times 10^{-3} xy + 1.644 x + 3.857 y + 171.6 \\ \dot{y} = 2.913 \times 10^{-4} y^2 + 1.674 \times 10^{-4} xy - 1.302 \times 10^{-3} x - 1.611 y - 4.622 \end{cases} \quad (1)$$

Now, we can study the stability of the system. We obtain numerically the critic points and discard those at negative, obtaining the following points. Hereinafter, we compute the eigenvalues of the Jacobian matrix evaluated at these points.

$$\text{Critical points : } (x_0, y_0) \in \{P_1 = (8018, 975), P_2 = (35010, 12)\}$$

$$\implies \text{The Jacobian } \mathcal{J} = \begin{pmatrix} -9.330 \times 10^{-5} x - 1.444 \times 10^{-3} y + 1.644 & -1.444 \times 10^{-3} x - 5.929 \times 10^{-3} y + 3.857 \\ 1.674 \times 10^{-4} y - 1.302 \times 10^{-3} & 1.674 \times 10^{-4} x + 5.827 \times 10^{-4} y - 1.611 \end{pmatrix}$$

$$\implies \text{Corresponding eigenvalues: } P_1 : \{\lambda_{11} = -0.107 + 1.422i, \lambda_{12} = -0.107 - 1.422i\}$$

$$P_2 : \{\lambda_{21} = -1.634, \lambda_{22} = 4.250\}$$

From these calculations, we can identify that the critical points belonging to the first quadrant are a center and a saddle point, respectively. Which is exactly what we see in Figure 12.

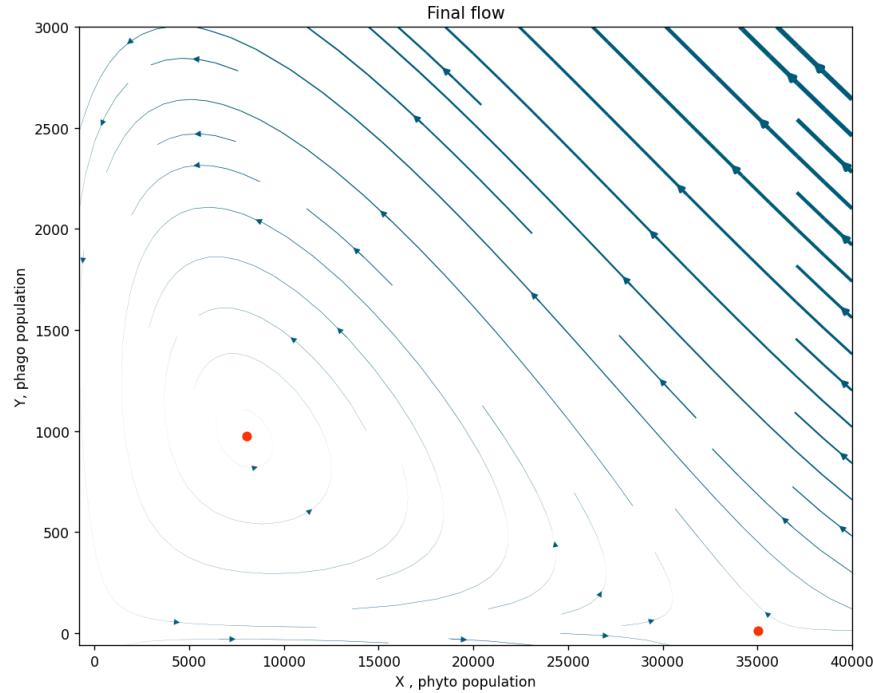


Figure 12: Vector field of found system with critic points marked.

It is particularly interesting, and it was certainly a surprise, that the SINDy was able to identify the saddle point, since it arises as a consequence of the system having a finite energy input and not by the oscillations of two species dynamics. The population of autotrophs grows until the carrying capacity is reached; this is the stable direction, and the unstable direction will be the one in which the population of heterotrophs increases.

When we calculate how well the model fits the training data, as we can see in Figure 13, is very high, with the exception of the extreme cases where a species begins to go extinct, similar to what happened before.

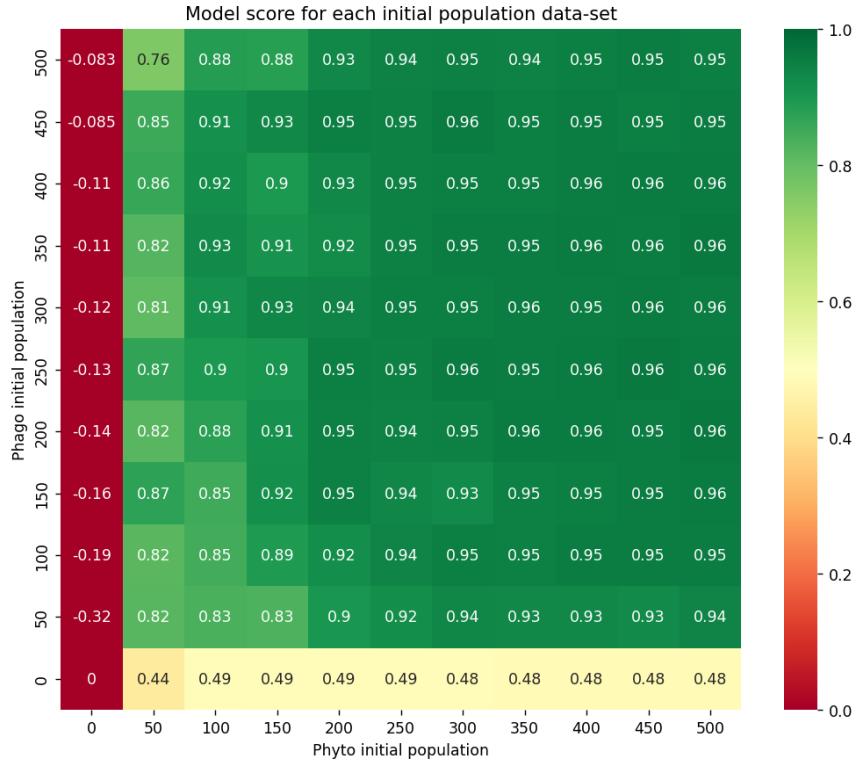


Figure 13: Score for the time derivative prediction produced by the model according to the initial population.

This gives enough evidence to show that the gradual decline of the oscillations is intrinsically present, but it does not give an explanation for why.

### 2.5.2 Comparison with Lotka-Volterra

As a quick comment, due to the sparse identification performed by SINDy, all the terms in the equation are important, even those that are accompanied by a very small coefficient in magnitude. Recall that the *phyto* population is at order values of  $1 \times 10^4$  and *phago* population at order of  $1 \times 10^3$ , so theirs square will be even larger. Therefore, the coefficients accompanying  $x^2$  and  $y^2$  are well chosen.

As a comparison, we can again apply SINDy, but this time limiting the library of functions to linear terms and  $xy$ , as it would be if the system followed a LV model.

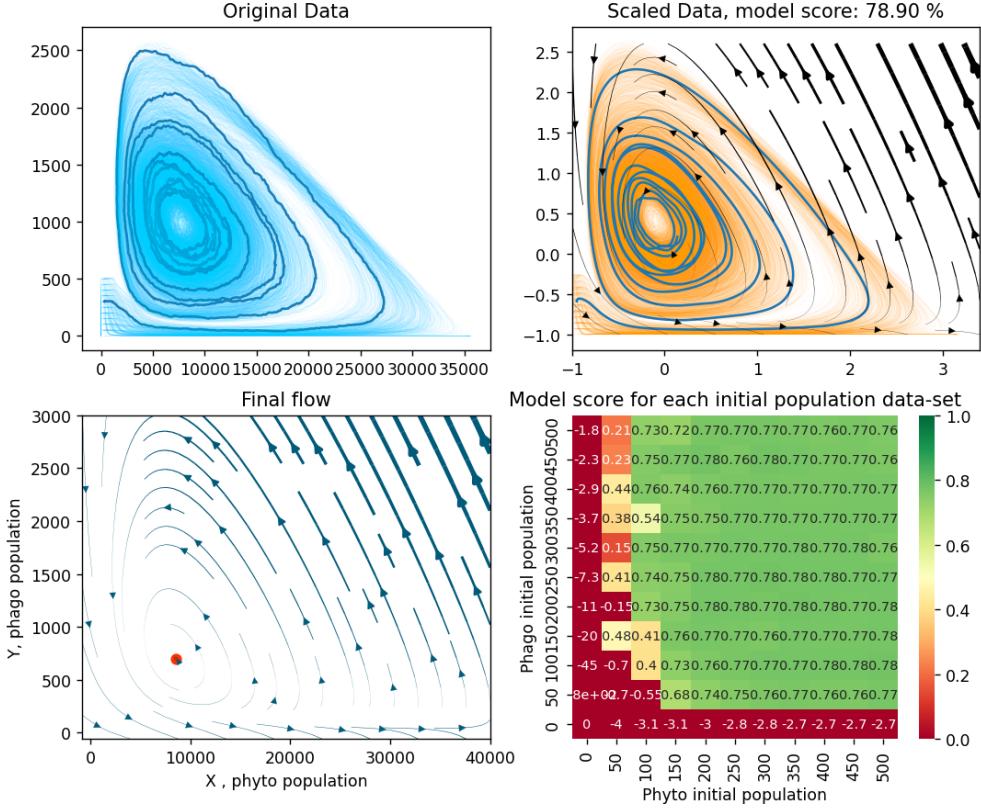


Figure 14: Results from forcing the SINDy library to not have quadratic terms.

Before your eyes, the found model does not include the dynamics of the data very well, in addition to the fact that the prediction score falls below 80%.

Lastly, it is true that the model found by SINDy has more cryptic constants than the LV parameters, which are interpretable with intuitive meaning. The study of the meaning of these constants requires other techniques (see subsection 6.3) and more work outside of this paper.

### 2.5.3 Cause hypothesis

Mathematically, the buffering effect is explained. But what is really happening in our simulation? The explanation has to do with the differences between microstates. For each macro-state (pair of populations *phyto-phago*), there are many possible spatial configurations. This has already been used to explain the discrepancies between maxima, but now more subtlety is required. The maxims change, but they follow a pattern.

When the simulation is initialized, all individuals take random position values. In general, this will assume that they are equispaced along the grid. Therefore, a *phago* will be able to prey less (it is

isolated from possible prey), while a *phyto* will find no competition for its autotrophic nutrition. This will result in much higher growth of the autotrophic population, resulting in a peak that results in most cases in the absolute maximum.

As the simulation progresses and new generations of predators and prey emerge, two parallel events condition population dynamics. On the one hand, once all the space is already used, competition between the *pytho* appears, and the *phago* will be able to prey more efficiently. In addition, the large size of the grid implies an uncoordination or lag in the ups and downs that decreases the amplitude of the "waves." It can occur simultaneously in two different parts of the simulation. The overlapping of partial dynamics disfavors a global dynamic. It may have a connection with the stochastic wave phenomenon.

## 2.6 Periodicity & frequency analysis

In this section, we study the quantitatively and qualitatively periodic properties of orbits and population waves.

### 2.6.1 Fourier analysis

Fourier analysis is a powerful mathematical tool used to decompose periodic functions into an infinite series of sinusoidal functions (sines and cosines) called harmonics.

The basic idea behind Fourier analysis is that any periodic function can be decomposed into an infinite sum of sinusoidal components with different amplitudes and frequencies. This is achieved by using the Fourier series, which represents the original function as a linear combination of sines and cosines of different frequencies.

This decomposition is useful because it allows us to analyze and understand the behavior of a function in terms of its simplest components. In addition, Fourier analysis has significant practical applications, such as data compression, signal filtering, sound synthesis, and differential equation solving.

We use this to better understand and look for patterns in the simulation dataset.

### 2.6.2 Frequency study of the data

Thanks to the eigenvalue calculation performed above, we can explicitly calculate what the period of oscillation around the center will be. To train the SINDy, we consider that an iteration corresponds to  $1 \times 10^{-2} \text{ s}$ .

$$\omega = \Im(\lambda) = 1.422 \text{ rad } s^{-1} = \frac{2\pi}{T} \implies T = \frac{2\pi}{1.422} \text{ s} = 4.418 \text{ s} = 442 \text{ iter}$$

Now with this information, we can perform a Fourier study to see how much each trajectory resonates at this frequency near the attractor, giving us a measure of how close the signals are to their limiting behavior.

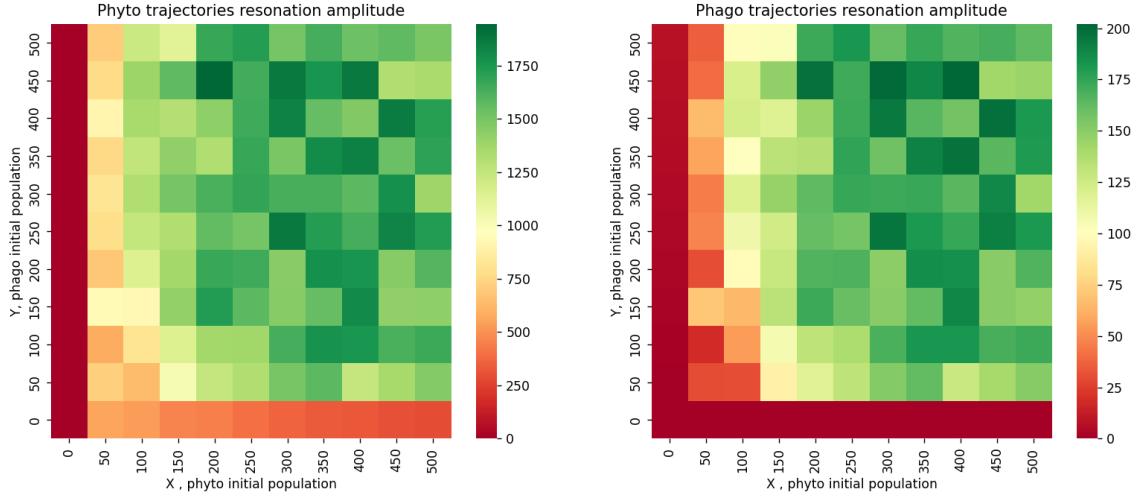


Figure 15: Modulus of the scalar product between trajectories and powers of the complex exponential for center frequency.

We can easily see in Figure 15 that higher the phyto and phago populations, the resonance increases. This is due to the fact that at higher populations, the system takes less time to approach the oscillating steady state. We can also clearly see that the dynamics that differ more (marked in orange) are farther away from the center, such as those close to the axes.

Furthermore, in the LV model, it happens that as you move away from the center point of equilibrium, you start to oscillate less frequently until you reach the limit of touching the axes, where you will never return, so you could say that the period will be infinite. In this sense, there is a negative relationship between amplitude and frequency. This plot leads us to the conclusion that the same thing happens in our identified system, with the distinction that we do not have closed orbits, so it is necessary to extend it for pseudoperiods and amplitudes.

In addition, for each initial population, we will study which frequencies predominate (they will be those that, for a set of simulations with the same initial populations, show the highest two-to-two coherence). Therefore, by iterating through all possible pairs of different simulations for the same initial conditions and obtaining an average, we can conclude the predominant components in the frequency spectrum.

Figure 16 shows the values of the maximum coherences for each initial condition. It allows us to deduce that for larger initial populations, the frequencies present in each simulation with the same initial conditions are more similar, thus seeing for which ones the result of the simulation is more predictable.

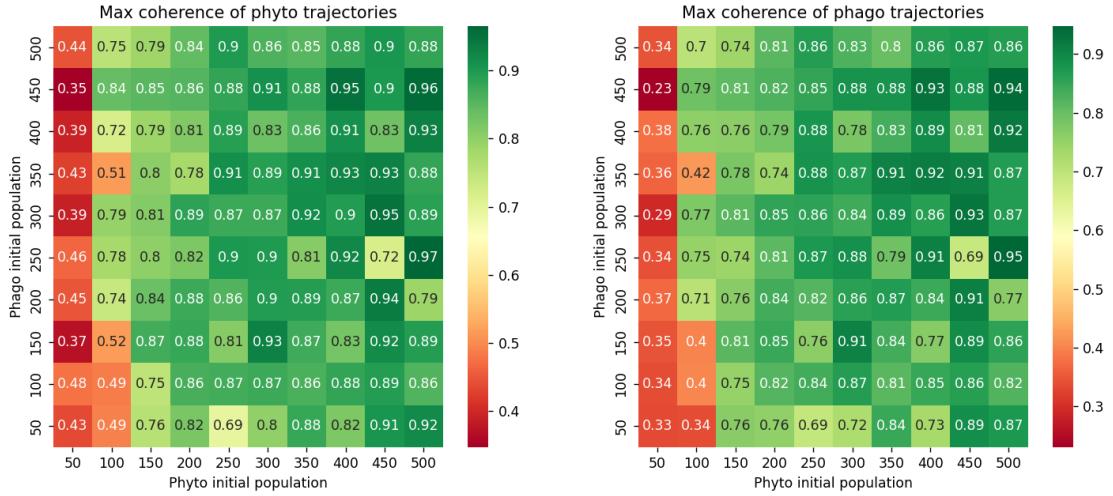


Figure 16: Frecuential coherences for each initial condition.

For this analysis, we have disregarded the data sets where one of the species started already extinct. As we will see in the next section, it is worth mentioning that, analogously to the other sections, the orange quadratics are not very representative since they have become extinct.

In any case, we see how the *phyto* and *phago* curves have the same frequency, and this increases slightly as we start with more *phyto*. Possibly because the transient state lasts less time and before we start spinning in the cycle.

## 2.7 Evaluation of model consistency and discrepancies with the LV model

By applying SINDy, we have managed to identify a system that fits better than LV with the data collected, but in return, we lose the intuitive interpretation of the system. In particular, the constant terms of the equations lead to a very different dynamic than LV.

We need to study a phenomenon of great ecological impact: extinction. In particular, in our simulation, for certain initial conditions, the total disappearance of heterotrophic individuals takes place. The behavior of ecosystems with only autotrophs is studied later (see subsection 3.1). In Figure 8, the orange colors (high discrepancies) correspond to simulations where extinction happened. As can be seen, all the situations in which extinction takes place coincide with a large number of initial heterotrophs and a small portion of autotrophs. These mass extinctions take place because a large population of predators needs a lot of energy, which it gets by devouring the few autotrophs that initially appear without giving them a chance to reproduce. Spatial arrangement again plays an important role (see section on buffering). It is sufficient for one *phyto* to survive for the population to regenerate.

On LV, when a species becomes extinct, it continues to be extinct indefinitely. This leads to

the fact that, dynamically, the four quadrants of the plane are not connected; there cannot be a trajectory that passes through any of the axes.

Our identified model does not comply with this. As a counterexample, it is easy to see that at the origin of coordinates, when both species are extinct, the velocity is non-zero.

This absurdity may have its origin in the crucial fact of spatial arrangement. As we have said, as an intuitive explanation, the local distribution of blobs favors the extinction of autotrophs, causing the velocity just prior to extinction to be non-zero.

On the other hand, it can also be caused simply and plainly by the fact that we are trying to study a discrete system by means of continuous equations. After all, in the simulation, the environment is updated discretely every iteration. We have already explained above that this segmentation causes the disparity of trajectories. As a reminder, when a species is on the verge of extinction (with two or three individuals), in some simulations it ends up extinct, and in others it thrives.

Phenomena that do not occur in the continuous case since the trajectory could continue to decrease asymptotically in a continuous and uninterrupted manner (just like in LV).

Considering that the plane is a grid of discrete points and that a particle can only move through contiguous points, it is possible to discretize the continuous system given by the equations.

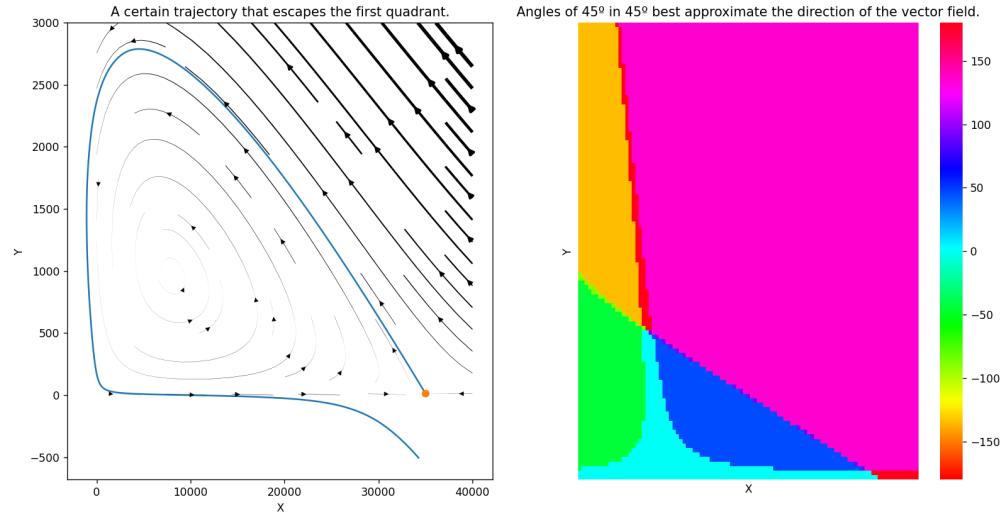


Figure 17: Comparison between continuous and discrete dynamics.

In Figure 17, we compare the same region of phase space, from the origin until  $(40000, 3000)$ . On display, the direction of flow to adjacent points is very close to the X-axis; the directions are either completely horizontal (to the right in cyan  $0^\circ$ ) or completely reversed (to the left in red  $180^\circ$ ). So it would not be possible to cross this axis since, in this region, the vertical velocity is zero.

### 3 Dynamics of one-species systems

This section studies the population dynamics of a single species (*phago* or *phyto*) in the absence of its complementary.

#### 3.1 Logistic growth and decay

Logistic population growth is a key concept in population ecology, offering a sophisticated model to understand how populations evolve over time within environments with finite resources. Unlike simplistic exponential growth models, which assume unlimited resources and unrestrained growth, the logistic model incorporates the crucial notion of carrying capacity.

At its core, the logistic model describes how population growth initially follows an exponential trajectory when resources are plentiful. However, as the population approaches the carrying capacity of its environment, factors such as competition for resources, predation, and limited space impose constraints on further growth. Consequently, the growth rate slows down until the population stabilizes at a level consistent with the carrying capacity.

Mathematically, the logistic equation captures this dynamic interplay between population growth and resource availability. It takes into account the intrinsic growth rate of the population (the rate at which it would grow in ideal conditions) and the carrying capacity of the environment. As the population approaches carrying capacity, the logistic function smoothly transitions from exponential growth to a plateau, reflecting the saturation of available resources.

Our simulation takes into account resource constraints. In fact, the way we regulate this phenomenon is by restricting the amount of energy a blob gains through its *phyto* parameter as a function of its environment. If the individual has potentially herbivorous neighbors, its energy gain will be lower. This is a way of introducing competition between entities.

For all these reasons, it is reasonable that, in the absence of predators, our simulation behaves like a logistic function as far as autotroph population growth is concerned. Indeed, this pattern is followed (see Figure 18).

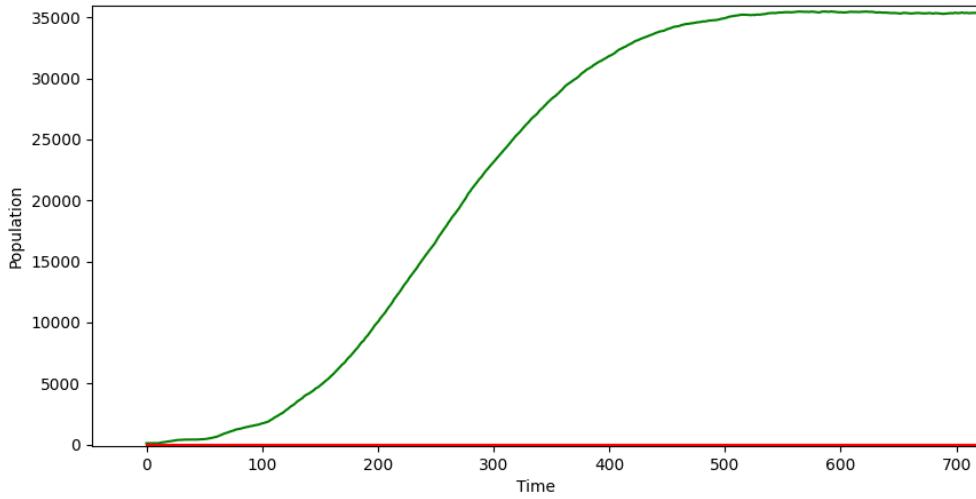


Figure 18: Sigmoidal population growth data

Having observed the phenomenon qualitatively, we would like to know exactly which logistic curve best approximates it. For this, we take the general form of a sigmoid:

$$\mathcal{S}(x) = \frac{c}{1 + e^{-a(x-b)}}$$

Which depends on three real parameters  $a, b, c$ . Now, we iterate through each of the files corresponding to population rises from 50 to 350 to reach the theoretical maximum number of individuals supported by the system. A state of thermodynamic equilibrium has been reached. We thus obtain the curves that best approximate our simulations (see Table 1).

Table 1: Optimized parameters to fit simulation data

Initial phyto	Parameters
50	$a = 0.0130; b = 320.26; c = 35288.65$
100	$a = 0.0163; b = 259.96; c = 35369.28$
150	$a = 0.0167; b = 237.41; c = 35307.26$
200	$a = 0.0172; b = 219.63; c = 35316.41$
250	$a = 0.0184; b = 203.98; c = 35311.54$
300	$a = 0.0183; b = 196.00; c = 35326.97$
350	$a = 0.0187; b = 187.30; c = 35307.57$

We can interpret the data in the table. The parameter  $b$  tells us the shift of the sigmoidal function to the right. Therefore, the higher  $b$  the less early the rise. We see that this agrees with what

we expect to see in the table: if the initial population is smaller, then they will be further apart and gain more energy from the environment; they will take up the energy to reproduce earlier, and therefore their shift to the right will be smaller.

The value  $a$ , on the other hand, indicates qualitatively the “rapidity” with which the function grows in the non-asymptotic interval (significant part). The greater  $a$ , the greater the slope. Therefore, as large initial populations start with a greater number of potential reproducers and the speed of growth in exponential behavior depends on the number of individuals at each moment, we expect a rise with a steeper slope and therefore a higher  $a$ , as shown in the table.

Finally,  $c$  is nothing more than the amplitude of the sigmoid and is always equal (beyond errors). It is identified with the maximum population capacity (about 35300 blobs).

### Extinction

Another phenomenon of great ecological interest is the behavior of extinction in the face of a lack of resources. Thanks to the tool, it is sufficient to impose a population composed only of heterotrophs. Interestingly, it follows a fall very similar again to the sigmoid (see Figure 19).

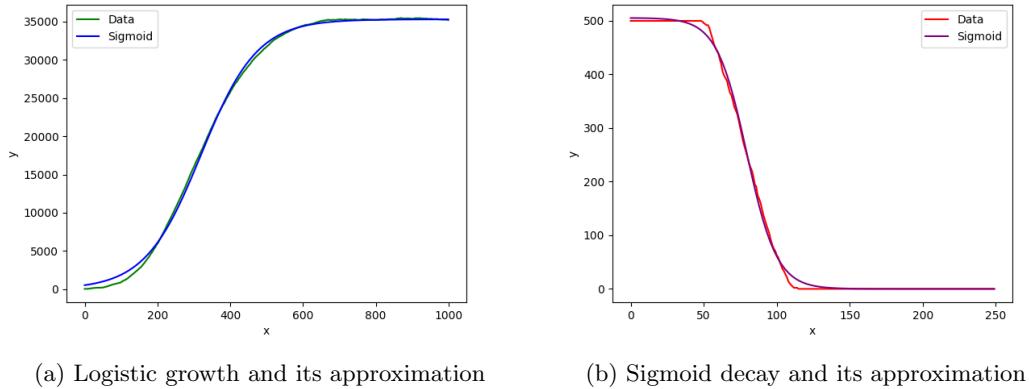


Figure 19: Sigmoid Approximation

## 4 Genetic adaptability and species emergence

The ability to adapt to overcome difficulties is the most impressive feature of nature. Our code takes this property into account by means of the variable *gen\_var*. This hyper-parameter (adjustable throughout the simulation by the users) gives us a unique opportunity to learn about how new species will emerge or are most suitable for the environment, and the rest of the individuals will be selected.

In this section, we will study the emergence and consolidation of species in ecosystems with minimum and maximum entropy.

## 4.1 Density-Based Spatial Clustering of Applications with Noise

DBSCAN, standing for Density-Based Spatial Clustering of Applications with Noise [9], is an algorithm employed for clustering data points within a dataset. It distinguishes clusters based on point density rather than predefined geometric shapes, rendering it adaptable and robust. By categorizing points into core, border, and noise points, DBSCAN effectively captures the intrinsic structure of the data. Its strength lies in its ability to handle noisy data and identify clusters of arbitrary shapes (see Figure 20).

We use the algorithm to find and count the species of blobs that emerge. A blob is nothing more than an array of values that encode their characteristics, thus defining a point in phase space. Points close to each other will form a cluster, which will be composed of those blobs with similar characteristics, thus defining species of blobs.

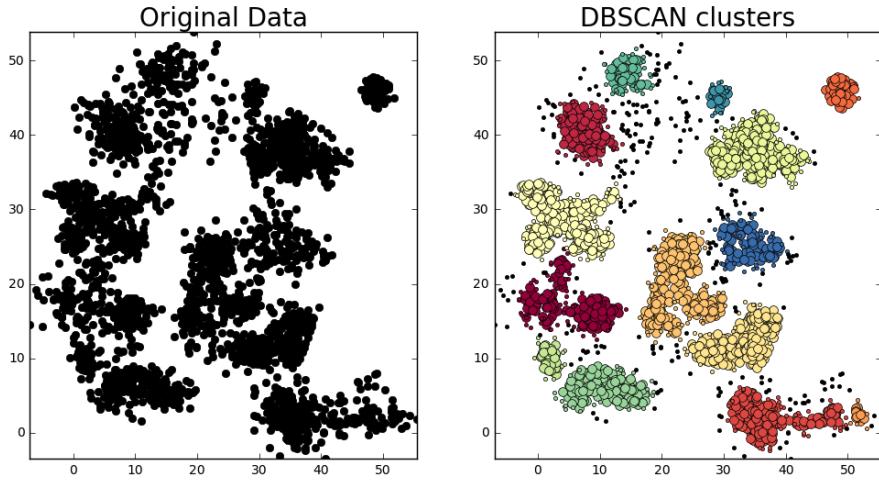


Figure 20: DBSCAN clustering example

## 4.2 Emergence of new species from Phyto

In this subsection, we discuss the evolution of an ecosystem initially composed of only autotrophs. Initially, we will have a minimum entropy (see Figure 21). Given the availability of resources and the existence of obvious ecological niches, the system will advance by exploring new possibilities thanks to the unpredictable factor of gene transmission.

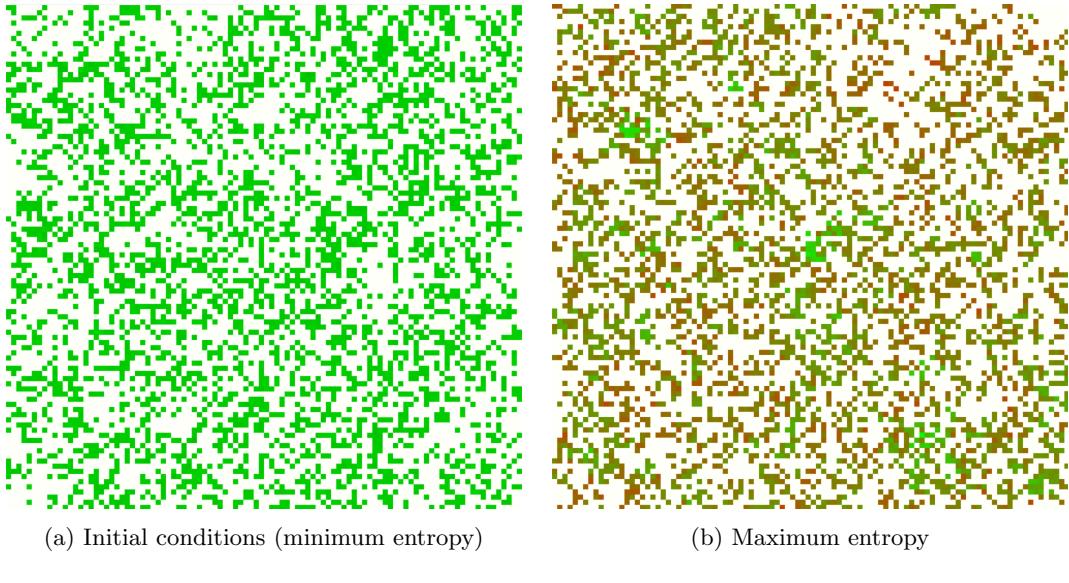


Figure 21: Increasing entropy

The reason for this lies in the direct application of the Second Principle of Thermodynamics: in an isolated system, its entropy always increases.

### 4.3 Genetic adaptability and ecosystem sanitation

This part discusses the natural selection phenomenon achieved in the simulation. We will start from an initial state of maximum entropy with individuals of random parameters. Thanks to genetic variability and competition, only the most adapted will survive and transmit their characteristics to the next generation. A system order is sought in which species emerge that occupy ecological niches. These species, once formed, evolve over time, mutually conditioning each other.

We will use a version of the ecosystem with a moderate *gen\_var* (see next subsection). We retrieve parameters such as *offens* or *defens*. More properties or parameters of the blobs allow for better separation and higher gene diversity, which is our goal in Sect.

We will employ PCA, followed by DBSCAN, to identify clusters representing species in phase space. We reduce the genes to four dimensions (the most significant ones), which we represent spatially and with the help of color.

The temporal evolution of a simulation in phase space is shown below.

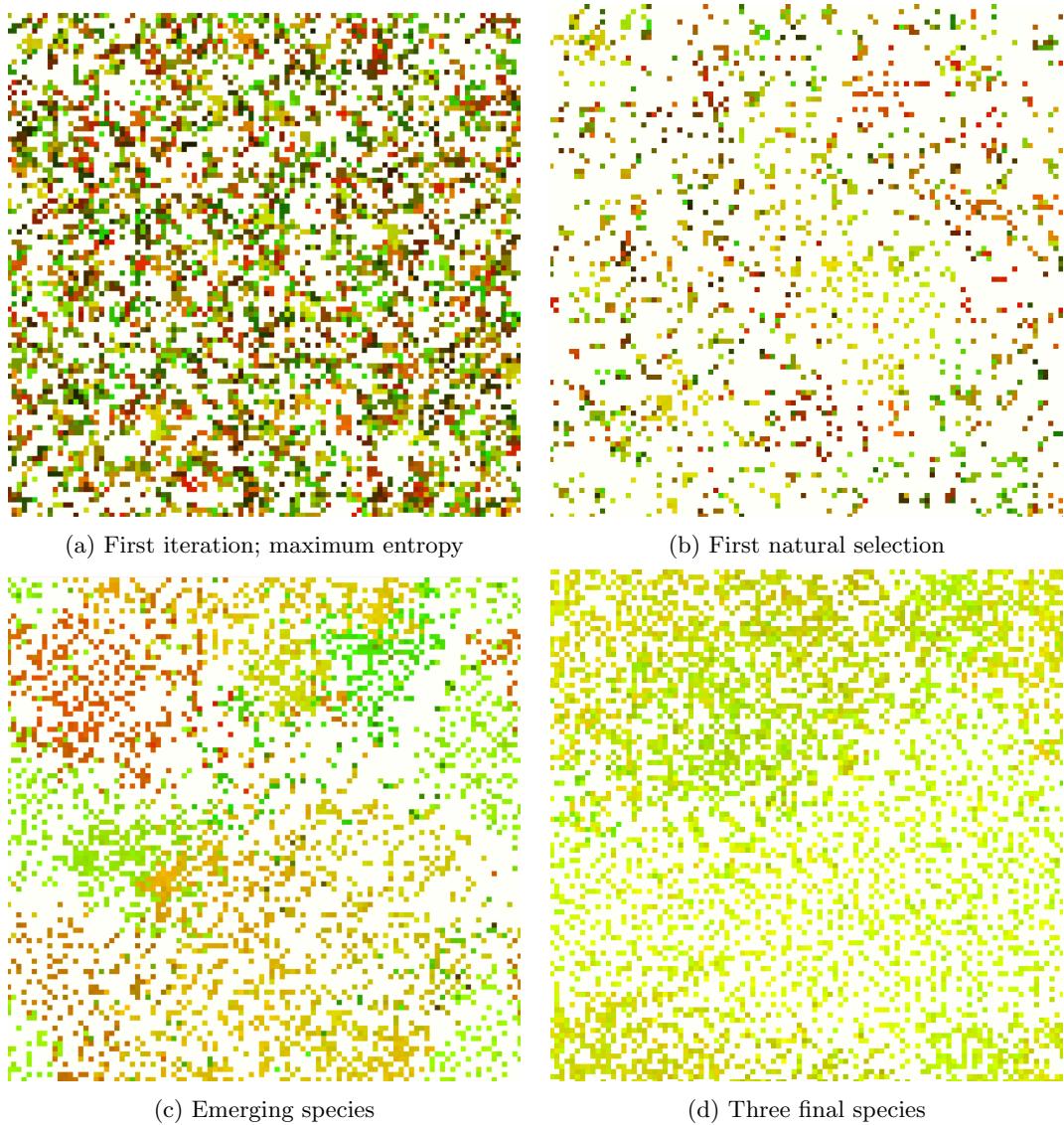


Figure 22: Temporal evolution of diversity

In these four images, we observe different stages through which the ecosystem passes. First, individuals with maximum variability are initialized (a). Due to the large number of non-adapted individuals, a first natural selection takes place, leaving a drop in population density (b). Later on, favorable ecological roles appear involving the formation of first species (c) that finally converge to the state that maximizes the obtaining of resources, three mainly autotrophic species with high levels of parameter *defens* that recover the population density with respect to the initial one (d). Thus, energy resources are optimized.

The following figure shows the results of the above simulation run through the PCA and DBSCAN methods.

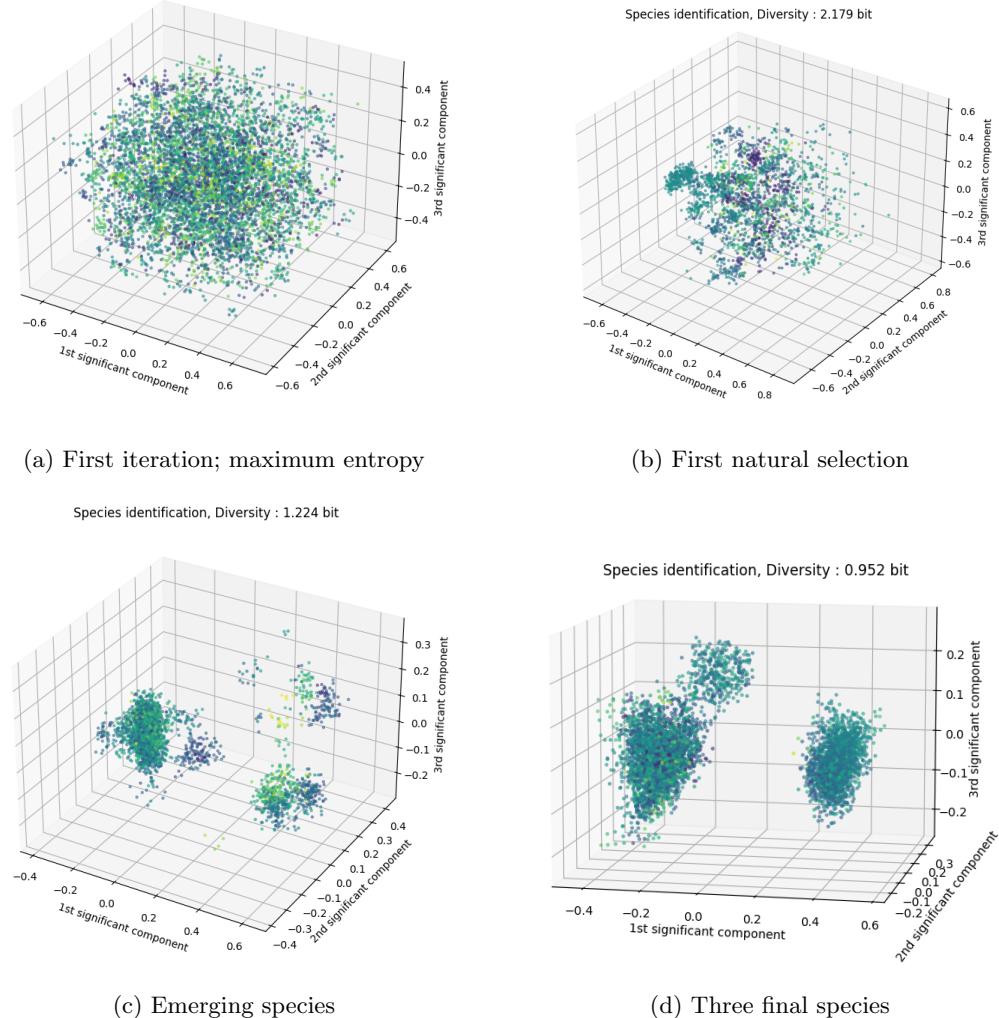


Figure 23: Temporal evolution of species (clustering)

We note that the axes are not always the same. Each time the clustering algorithm runs, the most significant components that change throughout the simulation are chosen.

In addition, performing an analysis of the information (entropy) of the system according to the species, we see how it clearly decreases (the first image has too much information and the method fails). This phenomenon of the purification of the ecosystem corresponds to the natural selection that we wanted to observe in this section.

#### 4.3.1 The importance of genetic variability

The value  $gen\_var$  is essential in the evolution and conformation of species (and thus of ecosystem dynamics). This is clearly seen by observing species formation for different values of the hyper-parameter with the same initial conditions (see Figure 24).

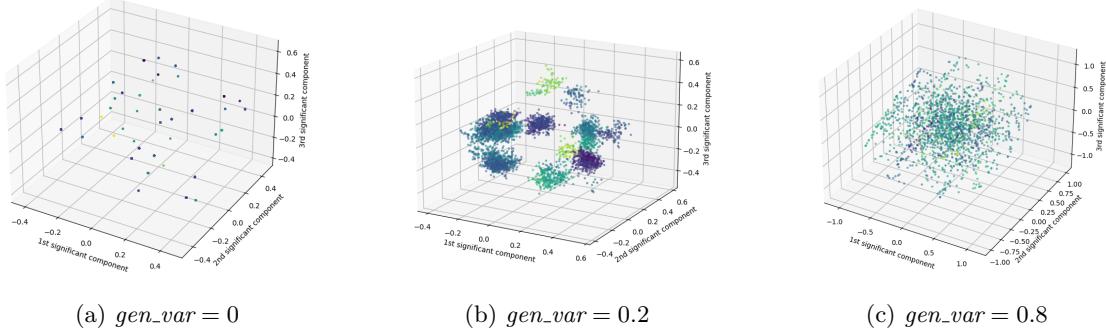


Figure 24: Importance of  $gen\_var$

## 5 Comments, difficulties, and observations

This chapter compiles the difficulties and challenges we have faced during the project. These are issues that, although already solved, we believe deserve a special mention for the solutions offered and what can be learned from them.

### 5.1 The complex predation balance

During the programming work, we have encountered a very frequent and difficult problem to solve. As we added new features to the genetic code of the creatures and endowed these improvements with a metabolic cost (as is natural), we saw that the role of predator was practically unfeasible.

The reality is that, in nature, the viability of heterotrophic nutrition is the result of a very long evolutionary process that would be difficult to replicate in a computer: a predator needs to invest energy in specializing its digestive system and improving its hunting strategies. The reflection of the former is the parameter of the final version called *phago*, while the latter is associated with the parameters *fight*, *vision*, *speed*, etc. And being a competitive organism in all these aspects is not energetically cheap and, therefore, hardly affordable. Moreover, when they appear in the zero iteration with random values, it is often not a coincidence that an individual appears with all these characteristics in a small favorable range and in a suitable position with respect to the prey.

The result of all these factors was homogeneous ecosystems in which very slow evolution limited the variety to a single herbivore species that very slightly evolved to converge to a completely passive omnivore.

Actually, there was a very straightforward solution. If we increased the amount of energy that carnivores were able to convert into their own, we would make this strategy more stable. However, this is very unclean and untrue to the reality we would want to simulate. At one step in the food chain, energy losses are estimated to be on the order of 90%.

So what did we do? Well, we noticed that the main problem was that evolution required many generations, each one of them assuming small changes that, although energetically costly, were not immediately advantageous. This had as a consequence (by natural selection) the difficult survival of these individuals promoting the autotrophic strategy. To solve this, we made small changes in the genetic code of parameters associated with the role of hunter to have more immediate repercussions.

Obviously, this is not a perfect solution. We also considered the option of setting initial values that would allow the immediate viability of the heterotrophic strategy, and this is, in fact, what we based the two-species version of the code discussed in previous sections on. We prefixed and simplified the qualities of the two species, looking for balance.

Personally, we have faced a real dichotomy between being faithful to the goal of allowing free evolution with little intervention and the need to replicate natural patterns. The ability of creatures to adapt to our mistakes and seek the most efficient way to take advantage of resources that are not always well distributed, helping us to improve the conditions of the ecosystem, is certainly to be appreciated.

## 5.2 Dependence on coordinates of SINDy

The SINDy algorithm has some assumptions, especially that it must be given an appropriately selected coordinate system and quality training data. Through the analysis in subsection 2.4, we consider the data to be of the highest quality we have achieved. This leads us to consider that the weak point of applying this method is the choice of the coordinate system.

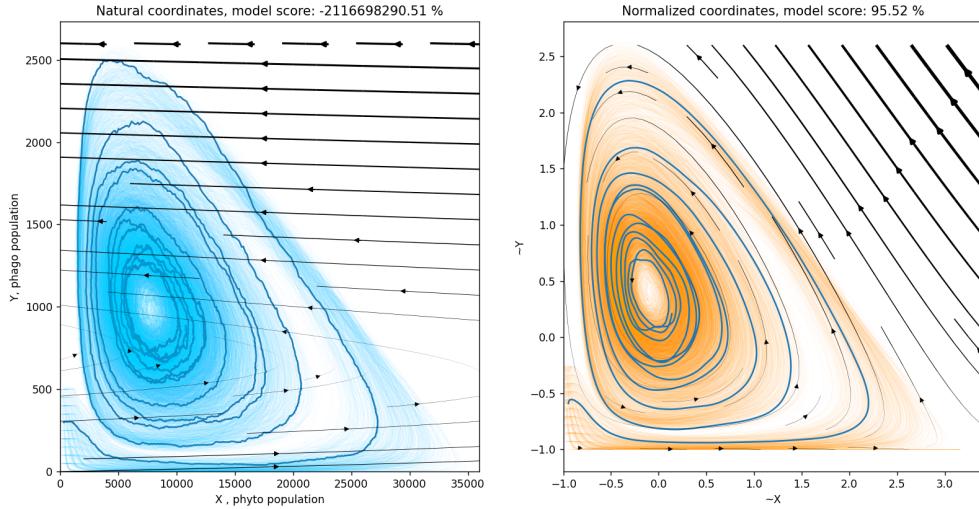


Figure 25: Comparison between applying SINDy with identical tuning parameters but with different coordinates.

In the process of exploring the library, we realized that working with the natural coordinate system, the X-axis representing the number of prey and the Y-axis representing the number of predators, gave us many sensitivity problems. See the comparison in the Figure 25. Therefore, in the pre-processing of the data, we have performed a normalization, since it is equivalent to performing an affine change of coordinates and clearly does the job very well.

Although the results are improved, this method has problems that have been previously exposed: the interpretation of constants and non-zero velocities in extinction (see subsubsection 2.5.2 and subsection 2.7).

The authors of SINDy, aware of this problem, have already provided solutions to it. In particular, they have published a SINDy autoencoder method for simultaneous discovery of coordinates and parsimonious dynamics[4].

In this work, we have made a proof of the concept of the methodology, and in a possible extension, it would be interesting to fully use the powerful tools provided by SINDy (see subsection 6.3).

### 5.3 Comments about the code

During the development of the code, certain features have been implemented to facilitate the user's interaction with the simulation.

We have started to write a manual, a series of files to navigate through that explain and give detailed information about the commands.

In connection with these commands, basic functions such as pause, accelerate, and decelerate the simulation speed have been implemented.

Since Pygame[6] is a library for making Python games, we have been able to easily implement a visual marking of the blobs, a color change to an eye-catching color that inherits downstream of the marked blob, allowing us to track individual blobs and observe their behavior.

Similarly, a method has also been implemented to visualize the complete list of physical characteristics of the blobs. This method is useful because, visually, only the food preference (*phyto-phage*) is represented by default, so to check other values quickly, print these values by terminal.

Lastly, and more importantly, it has been possible to modify the hyper-parameters interactively and play to change the system conditions when it has reached equilibrium and observe its evolution.

We leave [here](#) the link to our Github.

## 6 Future objectives

This section presents and explains code enhancements and studies that are already being done or will be done in the future.

### 6.1 Trophic web

One of the most interesting challenges we have for the future is to achieve the establishment of a trophic chain. This implies a more important advance than mere separation by species. We are looking for a specialization in predation. For this, a good tool is the parameter *fav\_meal*.

The *fav\_meal* feature of a blob stores the genetic code of its favorite prey. Whenever there is a predation interaction, it will be taken into account to determine the outcome and the energy gain of the winner.

Some of the problems we face are the correct balance of the impact of the *fav\_meal* on energy, its evolution over time, or the difficulty of reestablishing the predation balance mentioned in subsection 5.1.

An equilibrium can be found without great difficulty with a sufficiently large grid and a careful choice of the *fav\_meal*. However, the real difficulty lies in the spontaneous emergence of these stable trophic structures.

To recognize these structures, we associate each species (see DBSCAN) that is not clearly autotrophic with a favorite prey. Thus, we obtain a directed graph, which will be connected. Therefore, by identifying the herbivores, we will be able to sort them and thus identify their ecological roles.

These roles are organized hierarchically according to trophic level[1]. Therefore, given that in the simulation there are measurements of magnitudes that are physically very difficult to measure,

it is possible to calculate the energy present at each level of the network and thus perform a thermodynamic study of the system.

In addition, once the trophic network is identified as a graph, we can apply graph theory tools to study its connectivity, bottlenecks, resilience, and others.

## 6.2 Soil

We consider the richness of nutrients in the ecosystem to be an important factor. In the presented sections, the project does not consider any decomposition mechanics, but the reality is that this is something transcendental in ecological balances. Therefore, we are currently developing a version of the code that implements a new class called *Soil* that represents the nutrient quality associated with each cell.

This class is intrinsically linked to the grid and influences the energy that the autotrophic organism is able to obtain. It is, in simple terms, a matrix with entries associated with each cell with a real value between zero and one.

Throughout the iterations, these nutrients are distributed across adjacent cells, homogenizing the availability of resources. Furthermore, these values decrease as the organisms feed at their own expense and assume their maximum value when an individual dies in the cell in question.

This approach allows us to more accurately model how resources are distributed and consumed in the ecosystem. In addition, the dynamics of nutrient distribution between adjacent cells introduce an element of competition and cooperation between organisms. Those that are able to better adapt to variability in resource availability have a competitive advantage, which can influence ecosystem structure and evolution over time.

However, this increased complexity also brings challenges. Finding the right balance in nutrient distribution to avoid extinction or overpopulation of certain species becomes a non-trivial problem. It requires not only a thorough understanding of the underlying ecological processes but also the ability to translate that knowledge into simulation parameters that are interpretable and meaningful.

One ecological phenomenon that we have found very interesting is the cascade effect. This occurs when the loss or significant alteration of one species triggers a series of changes that propagate through multiple trophic levels. For example, the extinction of a predator species can lead to an uncontrolled increase in prey, which in turn affects the plants they consume and other species in the food chain. This phenomenon causes disturbances in the structure and function of the ecosystem, generating adjustments and adaptations throughout the biological community.

Once the chain is established and recognized, we can artificially eliminate a species to see the repercussions. This would allow us to understand more fully the problems caused by mass hunting or the loss of biodiversity.

### **6.3 Going deeper with the dynamic study**

We have used the SINDy tool for the first time on this job and have been pleasantly pleased with it. Due to this, we would like to see how far we can go with this tool.

The data set used in this work has been obtained for a particular hyper-parameter configuration. SINDy can identify dynamically parameterized systems. Thus, this tool gives us a quantitative method to study the relationships between the parameters of the environment and the dynamics of the system, thus opening many doors for us.

We can study the bifurcations of the system (the radical changes in the dynamics), depending on the energy input, genetic variability, and other hyper-parameters. We can look up expressions of the coefficients in terms of the equations as functions of these parameters, allowing us to bring them down to earth and look for interpretations of these parameters.

On the other hand, it would be interesting to try to find a coordinate system that favors the simplicity and interpretability of the system[4]. Since this would improve the quality of the model and the performance of dynamic identification,.

## **7 Conclusion**

The work embodied in this document is but a mere fraction of the dedication of the authors. The most valuable part of the project is undoubtedly the tool. To quote Abraham Lincoln, “Give me six hours to fell a tree, and I will spend four sharpening the axe.” This has practically been our philosophy.

As we progressed with the code, enriching the properties of the entities we had given (let’s say it was clear) life to, we got excited about the improvements we could offer. It has been, in short, a great personal enrichment.

We are tired, but proud of what we have achieved. The main objectives have been achieved, and new challenges have emerged that we will deal with in the future. We can openly say that it has been a resounding success.

## 8 Bibliography

Listed below are the texts from which key information or inspiration for this project has been drawn.

## References

- [1] Fionn Murtagh, Pedro Contreras. *Algorithms for hierarchical clustering: an overview*. December 2011.
- [2] Alfred J. Lotka. *Contribution to the Theory of Periodic Reactions*. 1910.
- [3] Brunton SL, Proctor JL, Kutz JN. *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*. March 2016.
- [4] Bethany Luschm, Steven L. Brunton, Kathleen Champion & J. Nathan Kutz. *Data-driven discovery of coordinates and governing equations*. October 2019.
- [5] Alan Garfinkel, Jane Shevtsov, Yina Guo. *Modeling Life*, Springer, 2017.
- [6] Pygame library [documentation](#).
- [7] Matplotlib library [documentation](#).
- [8] PySINDy library [documentation](#).
- [9] Scikit-learn library [documentation](#).
- [10] Atlassian Agile Methodology. [Atlassian Web](#).

## Academic documentation

- Volker Grimm, Steven F. Railsback & Christian E. Vincenot. *The ODD Protocol for Describing Agent-Based and Other Simulation Models*. March 2023.
- Luciano Stucchi, Juan Manuel Pastor, Javier García-Algarra & Javier Galeano. *A General Model of Population Dynamics Accounting for Multiple Kinds of Interaction*. July 2020.
- T. J. Clark & Angela D. Luis. *Nonlinear population dynamics are ubiquitous in animals*. January 2020.
- Richard Bischof, Cyril Milleret, Pierre Dupont. *Estimating and forecasting spatial population dynamics of apex predators using transnational genetic monitoring*. September 2020.

### 8.1 Acknowledgements

The authors would like to acknowledge the advice and collaboration of professors who have indirectly participated in the project by providing material and their expertise.

Thanks to Daniel López, Oscar Rodríguez, and Adrià Tauste, all of them from UPC.