

# A Simulation of the Extended Lotka-Volterra Systems for Natural Ecosystems Modeling

Project Collective Intelligence  
Group Joker

Olga P [REDACTED] - [REDACTED]  
Kayra Ö [REDACTED] - [REDACTED]  
M [REDACTED] Fatih A [REDACTED] - [REDACTED]  
Yusuf A [REDACTED] - [REDACTED]  
Sarp U [REDACTED] - [REDACTED]

**Abstract--***This study investigates whether stochastic, agent-based simulations can faithfully replicate stable, cyclic dynamics of predator-prey populations as outlined by the classical Lotka–Volterra (LV) equations and three discussed extensions. Three extensions to the standard LV model were implemented, optimized, and analyzed: (i) an LV variant with energy, food and spatial death mechanism factors, (ii) a Rock–Paper–Scissors three-species system, (iii) a three-agent E. Coli strain competition system, where agents are modeled as pixels to represent a microscopic environment. The results reveal that all of the implemented models consistently generated stable and robust cyclical dynamics. These findings support the hypothesis that incorporating ecological complexity doesn't necessarily negate the cyclic behavior and the stability of the simulated ecosystems.*

## 1 Introduction

### 1.1 Lotka-Volterra Model

The Lotka–Volterra (LV) model [1, 2] is a foundational mathematical framework in theoretical ecology, originally formulated independently by Alfred J. Lotka (1925) and Vito Volterra (1926). The framework focuses on deterministically describing the interaction between prey and predator species within natural ecosystems. The model describes gradual, interdependent changes in the population sizes of the two interacting species, where the system achieves stability understood as a coexistence of both species, with neither of them becoming extinct. Another distinct characteristic of an LV model is a cyclical pattern of population changes, which rises and decreases over time, creating sinusoid-like fluctuations.

### 1.2 Lotka-Volterra Model Extensions

Despite its ability to describe the changes in natural ecosystems, the LV model has long been criticised by researchers and ecologists for its limited realism [3, 4, 5]. The LV model is not well-suited for representing nuanced factors of the natural ecosystems, such as spatial heterogeneity or interactions between multiple

coexisting species [6]. For this reason, modern studies often extend the basic LV model by incorporating additional factors and creating a more complex theoretical framework for understanding natural phenomena [7, 8].

### 1.3 Limitations of the Theoretical Model

Although the extended LV models add essential dependencies of ecosystems, such frameworks are rarely converted into numerical simulations, which limits their practical utility. Translating theoretical models into simulations could allow researchers to experiment with the effect of additional features, which is not possible with deterministic, theoretical model descriptions or without disturbing real ecosystems. This could be particularly useful in cases where implementing such changes is impossible or unethical, for instance, when probing the ecological consequences of specific allele alterations [7]. The creation of efficient stochastic simulation methods could therefore provide researchers with a powerful tool that brings insights that deterministic analyses cannot supply. Thus, this research focuses on addressing the gap between theoretical formulations and practical implementation of extended Lotka–Volterra models.

### 1.4 Aim of the Study

This research aims to investigate whether stable, cyclic population dynamics of the Lotka-Volterra model and its extensions can be observed while theoretical models are converted into numerical simulations. For this purpose, a baseline Lotka–Volterra simulation model was developed alongside three extended variants, either self-designed or adapted from published deterministic models [7, 8]. The results were analyzed to see whether the simulations reproduce the stable, cyclic fluctuations in species' population sizes predicted by theoretical analyses. To evaluate this, the following hypotheses are proposed:

**H0:** In the simulated models:

- a) The cyclic behaviour will not be observed.
- b) Simulations will not create stable systems.

**H1:** In the simulated models:

- a) Cyclic behaviour will be observed.
- b) Stable ecosystems will be simulated.

## 2 Methodology

### 2.1 Simulation Environment

The models described in sections 2.2-2.4 were simulated using a Violet library [9] based on a PyGame engine [10]. The population behavior is modeled using agent-based modeling. Each species is defined as a distinct type of agent, characterized by its own behavioral logic or set of parameters. At every time step of the simulation, species population sizes are updated using fixed model parameters combined with rate functions that depend on the current population size and other simulation factors. Due to the limitations of the Violet library, the simulation of the E. Coli bacteria model (See 2.5) is implemented using the underlying PyGame engine [10], where each pixel represents either a bacterium or an empty space on a circular agar plate. All simulations were programmed using Python [11].

### 2.2 Baseline LV Model

**Classical LV equations** In the classical (baseline) Lotka-Volterra model [1, 2, 3], the predator-prey dynamics are described using the following derivations:

$$\begin{aligned}\frac{dN}{dt} &= \alpha N - \beta NP \\ \frac{dP}{dt} &= \delta NP - \gamma P\end{aligned}$$

where  $N$  and  $P$  denote, respectively, prey and predator populations sizes at a given time step  $t$ , where  $\alpha$  is the prey growth rate,  $\beta$  is the predation rate coefficient,  $\delta$  is the predator reproduction rate per prey eaten, and  $\gamma$  is the predator mortality rate.

**Simulation setting** In the implemented simulation, population size changes were established based on the structure of the classical differential equations, but the population sizes at the next time step were dependent on the interaction between the species (agents) rather than computed through derivation. No other spatial, energetic, or behavioural mechanics were added. It serves as a starting framework for implementing extended LV models.

The dependencies between prey and predator populations can be described as follows. In the absence of the predators, the prey population grows exponentially at a rate  $\alpha$ . The presence of predators decreases the prey population size at a rate  $\beta$  for each time step. The predator population size decreases at a constant rate  $\gamma$  when no prey is present in the ecosystem. The predator population reproduces probabilistically over consuming prey, at a rate determined by  $\delta$ . Furthermore, predator reproduction is dependent on the amount of prey consumed. The

given logic is implemented independently for each simulated agent, and population sizes at the next time step depend on the probabilistic outcome of the previous step.

### 2.3 Semi-Realistic LVM

The semi-realistic Lotka-Volterra simulated model is built based on the baseline model with agents' population sizes dependent on an additional factor of food energy dynamics. The energy factor affects both the mortality and reproductive rates of both species. The primary aim of the semi-realistic model is to depict prey-predator interactions in an environment that incorporates energy consumption, food dynamics, and spatial relationships between prey and predator populations, rather than simulating isolated reproduction and death of the baseline LV model.

In this model, each agent has an energy rate  $E$ , which decreases over time. The energy gained by consumption is a fixed parameter. The energy levels play a crucial role in the reproduction and survival of both prey and predator populations. Both need to maintain adequate energy levels to reproduce, and any agent whose energy level drops below zero will die.

**Prey energy mechanism** Prey agents can gain energy by consuming food available within their close radius. The food is introduced into the system at a constant spawning rate. The following equation can further describe the changes in the prey energy levels:

$$E_N(t+1) = E_N(t) - \epsilon + \sum_{f \in F_{near}} \eta * I$$

Where Energy of prey  $N$  at timestep  $t+1$  is determined by the energy decay rate  $\epsilon$ , the energy gain from food  $\eta$  and the presence of food  $I$  (1 if there is food present, 0 otherwise).

**Predator energy mechanism** Predators obtain energy by consuming their prey, as presented by the following equation:

$$E_N(t+1) = E_N(t) - \epsilon + \zeta * I$$

Energy of predator  $N$  at timestep  $t+1$ , where  $\epsilon$  is the energy decay rate,  $\zeta$  is the energy gain from prey, and  $I$  is 1 if there is prey present, 0 otherwise.

**Distance-based death** Predators don't need the prey to be within a close radius to consume it. Instead, at every timestep  $t$ , the probability of death of a given predator is dependent on its proximity to prey compared to that of other predators.

$$P_{death} = \gamma \left( 1 + \frac{dist_{nearest\ prey}}{100} \right)$$

Distance-based death probability for predators where  $\gamma$  is the base death rate and  $dist_{nearest\ prey}$  is the distance to the nearest prey.

### 2.4 Rock-Paper-Scissors Model

The Rock-Paper-Scissors (RPS) simulated model was created based on the theoretical framework as in [7].

The model extends the baseline LV framework by introducing a third species, with each of the species serving as prey to one of the other species and as a predator to another of the species. All three species move with a constant movement speed in the environment. Additionally, a self-crowding parameter is added, which moderates the otherwise exponential growth of the population sizes accordingly:

$$P_N(\text{death}) = \kappa \cdot N_x$$

Probability of death for an agent N is computed via the self-crowding coefficient and the population size of the species.

where the death probability of agent N at each time step is determined by the population size of the given agent species and the self-crowding coefficient  $\kappa$ , as the number of agents of the species increases, the probability of the agent's death rises accordingly.

Reproduction and predation mechanics follow the baseline probability mechanics, with a small perturbation ranging between -0.0001 and 0.0001 sampled from a uniform distribution added at every time step.

### 2.5 E. Coli Bacteria Model

The E. Coli bacteria model simulates the spatial dynamics of three competing strains engaged in a rock–paper–scissors-like (RPS) interaction as described in [8]. The three species follow an asymmetric competitive cycle, in which Strain R (Red) dominates G (Green), G dominates B (Blue), and B dominates R. These strains correspond to the production of Colicin E3, Colicin V, and Colicin E7 toxins, respectively. However, the model abstracts these into probabilistic domination rules.

The bacteria are initialized in a grid pattern, like in the original paper. Each bacterial strain can reproduce into adjacent empty pixels with a probability determined by the reproduction rate ( $\alpha$ ). The death rate ( $\mu$ ) introduces random cell death at each frame if the cell is not empty. Bacterial strains can also dominate competing strains with a probability governed by the domination rate ( $\delta$ ), scaled by a fixed toxin strength ( $T_i$ ) as shown in the equation.

$$\text{domination death} = \delta \cdot \frac{T_i}{T_{\text{weakest}}}$$

An occupied cell has a domination death probability computed with the death rate and scaled by toxin strength.

### 2.5 Statistical Analysis

The statistical analysis examined the presence of cyclical patterns and assessed the long-term stability of the simulations. Model cyclicalness was defined as the occurrence of sinusoid-like fluctuations in population sizes over time, quantified using the Augmented Dickey-Fuller (ADF) statistical significance test [12]. Model stability, understood as

the coexistence of two species without either becoming extinct during the simulation, was operationalized as the absence of diverging trends in the population means, evaluated using the analytic-signal approach (AS) [13]. Statistical analyses were conducted using Python [11], with statsmodels and SciPy libraries [14, 15].

For each model, the simulation was repeated 50 times. The stability and the cyclicalness of the changes in agent population sizes were then assessed independently for each run. Because both properties are influenced by stochastic variation and can differ across repetitions, an operational threshold was defined: a simulated model exhibits a cyclical and stable behavior if both of these behaviors are observed simultaneously in at least 12% of runs (i.e., 6 out of 50). This criterion ensures that the likelihood of observing such a result is not attributable solely to random false positives arising from repeated statistical tests. The threshold reflects the exploratory nature of the study, which is not intended to produce highly reliable systems. Instead, this research aims to assess whether recreating stable and cyclical dynamics is feasible in a stochastic simulation environment.

## 3 Experiments and Results

### 3.2 Experimental Setup

The simulations of all four models were initialized (Fig. 1.) without a fixed seed for the simulation. All simulations had a maximum duration of 5000 frames.

Model	Parameter	Species A	Species B	Species C
Baseline	Initialization count	100 (prey)	20 (predator)	-
Semi-Realistic	Initialization count	100 (prey)	20 (predator)	-
RPS	Initialization count	30	24	6
E. Coli	Toxin strength	4.17	1.53	1.0

Fig. 1. Initialization parameters of all models

### 3.3 Hyperparameter Optimization

Due to the large number of parameters in each model, the exact values of each of the parameters described in the Methodology section were optimized using grid search and evolutionary optimization techniques.

**Grid Search** All parameters of the baseline model and the semi-realistic model, i.e., initial population sizes, birth and mortality/ predation rates, and energy decay rate, were optimized using a grid search, with parameters seeded from the original LVM paper and

energy parameters randomly selected within the ranges between 0.01 and 3.

**Evolutionary Optimization** The parameters of the simulations of models 2.4 and 2.5 were optimized using evolutionary optimization techniques due to the considerable increase in the number of simulation parameters and therefore the search space. Grid search lacks the complexity and adaptability, whereas evolutionary methods offer a robust, population-based approach capable of navigating high-dimensional and noisy search spaces, making them well-suited for tuning the dynamic behaviors observed in these simulations. [16, 17]

**Rock-Paper-Scissors Model** In the RPS model, the fitness function was selected to be a Fast Fourier Transform (FFT). FFT is an efficient algorithm for computing the Discrete Fourier transform (DFT), which converts a time-series signal into its frequency components, revealing periodic patterns in population data [18, 19]. The resulting fitness score quantifies the strength of oscillatory behavior across the simulation. The mutation method used was Gaussian (Normal) mutation, and the crossover method selected was Uniform crossover. The algorithm also employed an elitist approach, ensuring that mutations occurred on top of the best results [16, 17]. The parameters optimized were the birth rate ( $\alpha$ ), predation rate ( $\beta$ ), crowding coefficient ( $\kappa$ ), and movement speed.

**E. Coli Bacteria Model** In the E. Coli model, the fitness function was defined as a score computed by multiplying the simulation duration by the negation of the standard deviation rate over the mean at the final time step of the simulation. The mutation method used was uniform re-sampling, where each gene is replaced with a new random value based on a given probability. This algorithm did not contain a crossover method. The survivorship selection was performed using tournament selection, and an elitist approach was employed for parent selection. Only the simulation parameters of death rate ( $\mu$ ), reproduction rate ( $\alpha$ ), and domination rate ( $\delta$ ) were optimized. In contrast, the toxin strength of each species was kept as outlined in the paper describing the E. Coli model [8]

### 3.3 Baseline Model Results

The results of the baseline LV model (Fig. 2.) indicate that the simulation has reached and exceeded the set threshold of at least 12% cyclical and stable simulations. Out of 50 runs, 36 (72%) presented both stable and cyclical simulation runs. The same number of simulations has reached a stable solution, *ADF stat* range of significant results range: [-26.52, -3.07], with  $p < 0.029$  or smaller for all significantly stable runs. Cyclical patterns were observed in 49 runs (98%) with an AS  $\omega$  range of statistically

significant solutions:  $[-0.008, 0.275]$ . These results support the assumption that the LV model can be simulated using stochastic simulation.

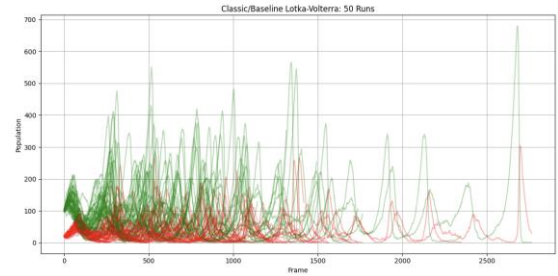


Fig. 2. Population dynamics in the baseline LVM model throughout 50 runs. Green line shows prey population, red line shows predator population.

### 3.4 Semi-Realistic LV Results

The results of the semi-realistic model simulation (Fig. 3.) also reached the threshold of 6 (12%) stable and cyclical runs, with both stability and cyclicity found in 86% of the cases. Statistically significant cyclic behavior was observed in 96% of the cases, AS  $\omega$  range of statistically significant solutions:

$[-0.174, 0.228]$ . The system was found to be stable in 90% of the cases, *ADF stat* range of significant results: [-19.73, -2.99]  $p < 0.03$ . This result confirms the presence of stable and cyclical runs within the simulated semi-realistic LV model.

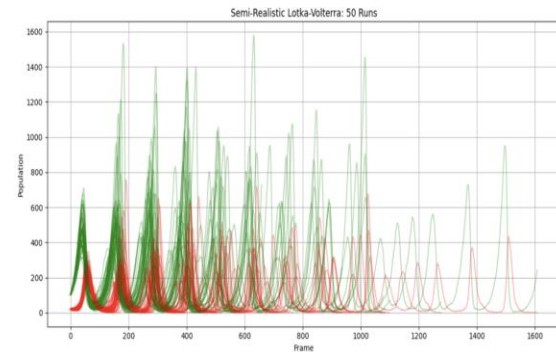


Fig. 3. Population dynamics in the semi-realistic LVM model throughout 50 runs.

### 3.5 Rock-Paper-Scissors Model Results

The results of the RPS model (Fig. 4.) exceed the success criterion of six stable-cyclical runs (12%): 47 out of 50 replications (94%) test significantly for both the stationarity and phase-drift tests simultaneously. Cyclicity test alone was significant in 47 of 50 runs (96%), with the  $\omega$  score ranging between  $[-0.174, 0.228]$ . Stability was found in 96% of the runs, with the significant *ADF stat* range of [19.73, -2.99] and the highest significant  $p$  value of 0.0062. These results demonstrate that the RPS

model accurately reproduces the predator-prey cycle in stochastic replications, with strong statistical support for both cyclicity and stability.

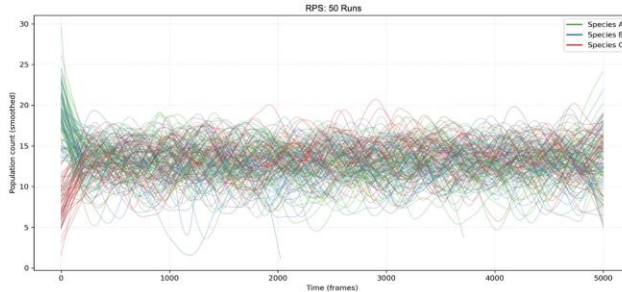


Fig. 4. Population dynamics of RPS model over time, 50 runs.

### 3.6 E. Coli Model Results

The results of the E. Coli model (Fig. 5.) has also reached the threshold of 6 (12%) stable and cyclical runs, with both stability and cyclicity observed in all 50 runs (100%) of the cases, AS  $\omega$  range of statistically significant solutions:  $[-0.066, 0.014]$  and ADF  $stat$  range of significant results:  $[-7.30, -4.59]$   $p < 0.001$  for all of the runs. This result suggests that non-transitive, rock-paper-scissors interactions create stable and cyclic multi-species dynamics.

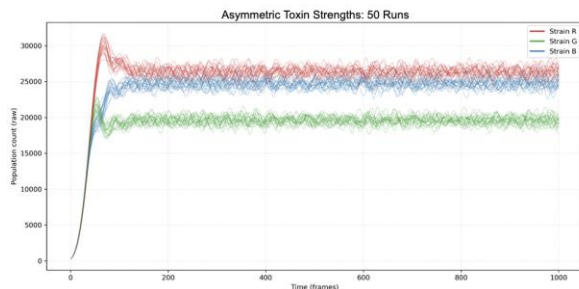


Fig. 5. Population dynamics of the bacteria model throughout 50 runs.

### 3.7. Interpretation

The analysis of the simulated models' behavior has confirmed the presence of cyclical patterns and model stability under the implemented simulations, thereby providing evidence to reject the null hypothesis that cyclical behavior and stable model behavior will not be observed. All of the created models have exceeded the operational threshold of stable and cyclical patterns, with the percentage of both stable and cyclical runs ranging between 76% and 100% of the simulated runs, providing evidence in favour of the alternative hypotheses H1a and H1b. This indicates that the simulation of the LV model and its extensions will exhibit cyclic and stable dynamics.

One crucial observation is that, despite the baseline and semi-realistic models' high rates of statistical stability, the prey and predator populations within their simulations have always collapsed to extinction before the simulation was completed. This discrepancy highlights an apparent mismatch between the formal test results and the actual long-term behaviour of the system (see Section 4.2).

## 4 Conclusion

### 4.1 Findings

This study investigated whether stochastic simulations of theoretically extended Lotka–Volterra models can generate stable, cyclical predator–prey dynamics predicted by their theoretical, deterministic formulations. The presence of cyclical ecosystems was characterized by sinusoid-like fluctuations in the population sizes of the model. System stability was characterised as a situation where no species goes extinct over time. The presence of both traits was found simultaneously in all simulated models. Therefore, it can be concluded that the extended LV models can be simulated using stochastic simulations, with the characteristic patterns of prey and predator dynamics preserved.

### 4.2 Limitations

The research has several limitations arising from the chosen testing methods, the simulation environment, and the structural assumptions of the models under investigation.

Although all the models were assessed to have high stability rates, the behavior and statistical results of the baseline and semi-realistic models appear to be misaligned, with the species always going extinct before the end of the simulation. The apparent disparity may have several possible explanations. Firstly, it is possible that the statistical method used to assess model stability may not have been sensitive enough to accurately measure the model's actual behavior. This could have caused the model to be falsely categorised as stable. Alternatively, the result might have been dependent on the number of prey and predators in the created simulations. With the current fluctuations in population sizes, the number of agents of certain species often drops to a single individual. In this moment of a simulation, a small perturbation to the general fluctuating population patterns can cause the species to become extinct. Such fluctuations are inevitable in a stochastic, created environment. The result might therefore be an unavoidable outcome of the current parameter setting of the simulations.

Another important limitation arises from the complex parameter dependencies inherent in the created models. In these multi-parameter simulations, interactions between parameters are highly nonlinear



and difficult to evaluate systematically. Exhaustive exploration of the parameter space is unfeasible, as even small changes in parameter values often result in large and unpredictable shifts in system behavior. This complexity makes optimization challenging and limits the comparability of different simulation runs. Moreover, given the vast landscape of possible parameter combinations, the current study cannot rule out that other, unexplored configurations could produce more stable or longer-lasting solutions. Consequently, the findings only reflect the specific parameter sets tested, leaving many potentially relevant outcomes unexamined.

A further limitation of the simulation process stems from the simulation environment itself. During the research, the behaviour of the E. Coli model could not be accurately reproduced within the Violet [9] framework, due to mismatches between the simulator's architecture and the internal logic of the implemented model. This suggests that similar incompatibilities could arise in other applications, potentially limiting the validity of results. Accordingly, the design and assumptions of the simulation environment should be carefully reviewed to ensure they align with the structural requirements of each model.

The chosen simulation environment should also be considered as a factor limiting the internal validity of the presented results. There is a possibility that the observed dynamics might be partially dependent on the simulation code, rather than solely on the implemented behavioral logic of the models' agents. This issue should be considered when utilising similar simulations to study the underlying ecological dynamic systems.

### 4.3 Future Improvements

Throughout the model design phase of this study, several neuroevolutionary approaches [20] were explored (i.e. pre-training, linear interpolation between PFSM logic and learned NN behavior [21], co-evolution amongst prey/predators [22], self-adaptive mutation via score matrices [23], etc.) to construct a hyper-realistic ecosystem in which agents could predict their next movement and internal state at each simulation frame based on a training phase. However, a key limitation emerged in terms of computational feasibility: the training phases for each species often exceeded 24 hours, rendering the approach impractical within the constraints of the available hardware and project timeline, and therefore the model infeasible. Further studies equipped with greater computational resources and time could

explore this approach in more depth and design a hyper-realistic model, in the sense of agent consciousness.

## 5 References

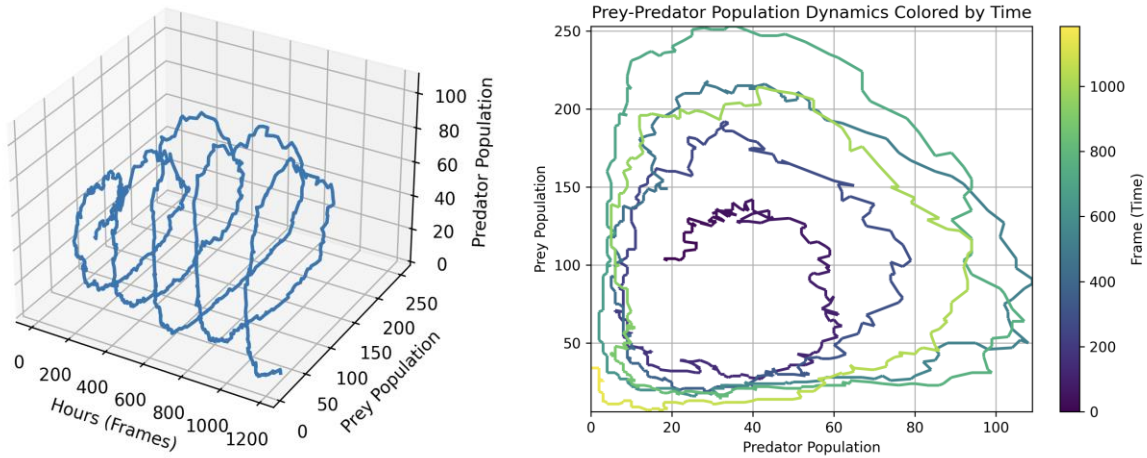
- [1] A. J. Lotka, Elements of Physical Biology. Baltimore, MD, USA: Williams & Wilkins, 1925.
- [2] V. Volterra, "Fluctuations in the abundance of a species considered mathematically," *Nature*, vol. 118, no. 2972, pp. 558–560, 1926. [Online]. Available: <https://doi.org/10.1038/118558a0>
- [3] Maas, Han. (2024). Complex-Systems Research in Psychology. 10.37911/9781947864634.
- [4] Jensen, H. J. (2022). Complexity science: the study of emergence. Cambridge University Press
- [5] Shen, H., Zhu, Y., Chen, H., & Zhang, D. (2015, June). An bio-inspired complex system dynamic optimization decision method. In 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER) (pp. 121–126). IEEE.
- [6] Wangersky, P. J. (1978). Lotka-Volterra Population Models. *Annual Review of Ecology and Systematics*, 9, 189–218. <http://www.jstor.org/stable/2096748>
- [7] Barendregt, N. W., & Thomas, P. J. (2023). Heteroclinic cycling and extinction in May–Leonard models with demographic stochasticity. *Journal of Mathematical Biology*, 86(2), 30.
- [8] Liao, M. J., Miano, A., Nguyen, C. B., Chao, L., & Hasty, J. (2020). Survival of the weakest in non-transitive asymmetric interactions among strains of E. coli. *Nature Communications*, 11, 6055. <https://doi.org/10.1038/s41467-020-19814-w>
- [9] Timmerman, S., "Violet 'vi' Simulation API," <https://api.violet-m-rots.com/vi?ref=canvas>
- [10] Pygame Community, "Pygame: Python Game Development Library, ver. 2.6.1," 29 Sep 2024. [Online]. Available: <https://pypi.org/project/pygame/>
- [11] Python Software Foundation, "Python Programming Language, ver. 3.12.0," 02 Oct 2023. [Online]. Available: <https://www.python.org/>. [Accessed: 29-06-2025].
- [12] K. S. Said and D. A. Dickey, "Testing for unit roots in autoregressive-moving average models of unknown order," *Biometrika*, vol. 71, no. 3, pp. 599–607, 1984.
- [13] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal. I. Fundamentals," *Proc. IEEE*, vol. 80, no. 4, pp. 520–538, Apr. 1992.
- [14] S. Seabold and J. Perktold, "statsmodels: Econometric and statistical modeling with Python," in *Proc. 9th Python in Science Conf. (SciPy 2010)*, Austin, TX, USA, 28 Jun.–3 Jul. 2010, pp. 57–61.
- [15] P. Virtanen et al., "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nat. Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2020.
- [16] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, 1996.
- [17] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, 2nd ed. Springer, 2015.
- [18] D. V. Fedorov, "Fast Fourier transform (FFT)," lecture notes, Dept. of Physics and Astronomy, Aarhus University, Denmark. [Online]. Available: <https://phys.au.dk/~fedorov/Numeric/11/fft.pdf>
- [19] S. Queiroz, "Fast Compressed-Domain N-Point Discrete Fourier Transform: The 'Twiddless' FFT Algorithm," arXiv, May 29 2025. [Online]. Available: <https://arxiv.org/abs/2505.2371>
- [20] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002. [Online]. Available: <https://nn.cs.utexas.edu/downloads/papers/stanley/ec02.pdf>
- [21] J. Lucas and P.-W. Verbeek, "Linear path neuroevolution for continuous control," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, vol. 139, 2021. [Online]. Available: <https://proceedings.mlr.press/v139/lucas21a.html>
- [22] F. Gómez, "Accelerated Neural Evolution through Cooperatively Coevolved Agents," Ph.D. dissertation, IDSIA, Lugano, Switzerland, 2008. [Online]. Available: <https://people.idsia.ch/~juergen/gomez08a.pdf>
- [23] Rajendran, R., Nath, A., & Ghosh, A. (2024). Modeling bio-inspired swarm robotics with neuroevolution and self-organization. *PLOS ONE*, 19(6), e0307084. Available: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0307084>

GitHub repository:  
[https://github.com/OlgaPiestrak/Collective\\_Intelligence\\_Project](https://github.com/OlgaPiestrak/Collective_Intelligence_Project)

## 6 Appendix

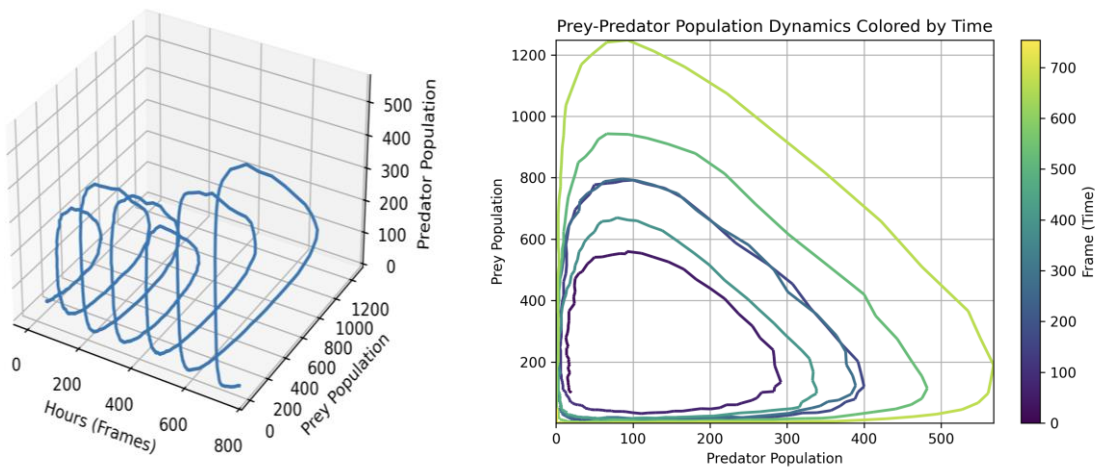
### *Extra 3D plots of baseline and semi-realistic models from presentation 1*

Prey-Predator Dynamics with Time

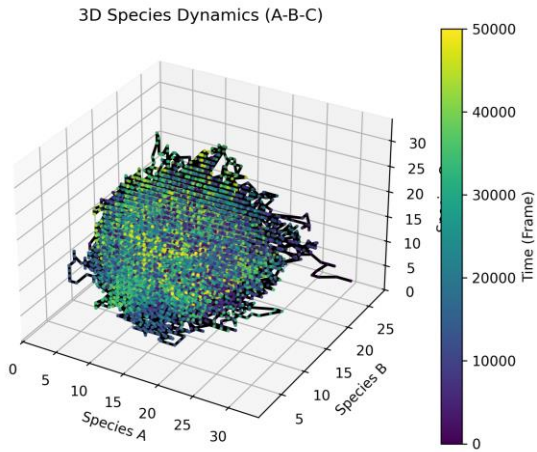


*Baseline Lotka-Volterra model population dynamics over time & population dynamics colored by time*

Prey-Predator Dynamics with Time



*Semi-realistic Lotka-Volterra model: population dynamics over time & population dynamics colored by time*

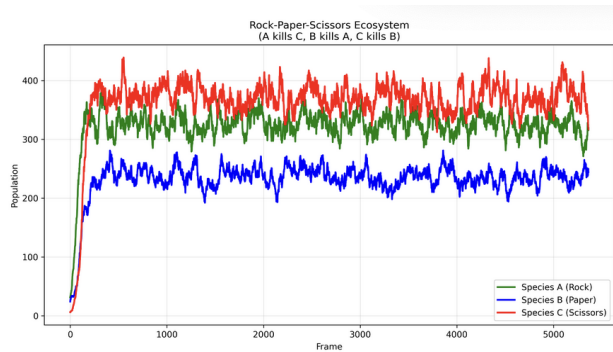


*Rock-Paper-Scissors Lotka-Volterra model population dynamics colored by time (this graph was not used in any presentations)*

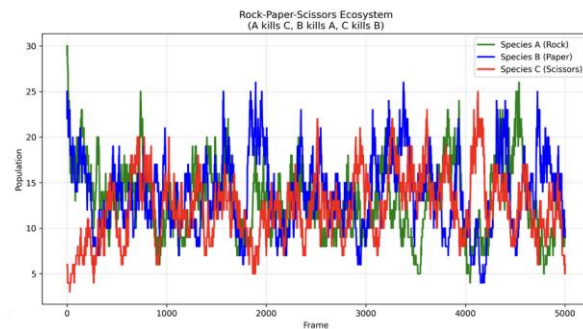


**Optimization Differences:** Optimization differences plot (the optimized parameters were used directly during statistical testing, the difference between non-optimized and optimized parameters can be seen below):

### Rock-Paper-Scissors model implementation

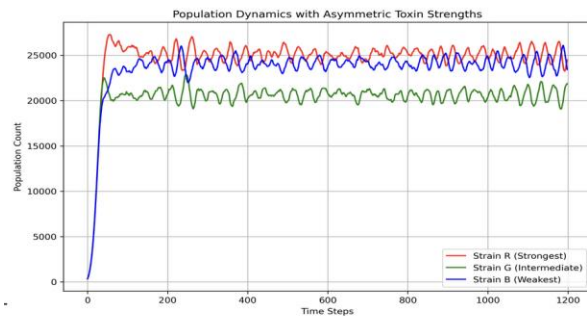


*Results of the model with parameters optimized via evolutionary methods*

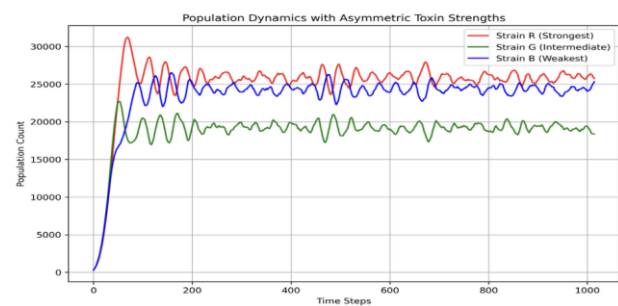


*Results of the non-optimized parameters directly adapted from the paper.*

### Bacteria model implementation

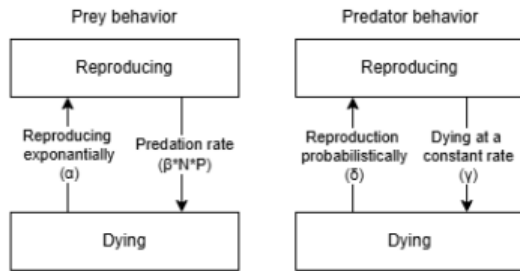


*Results of the simulation with optimized parameters via evolutionary methods*

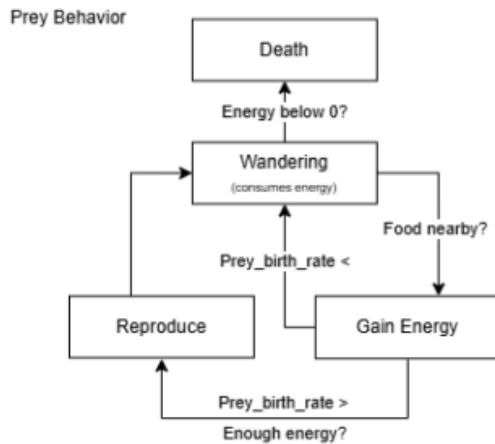


*Results of the simulation with parameters from the original paper*

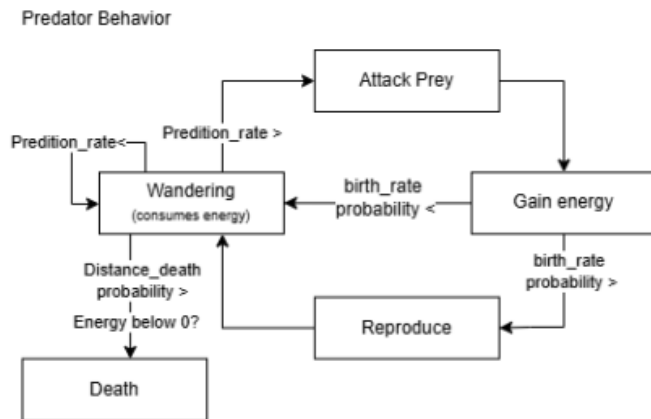
*Flowcharts used in the presentation sessions to explain the models:*



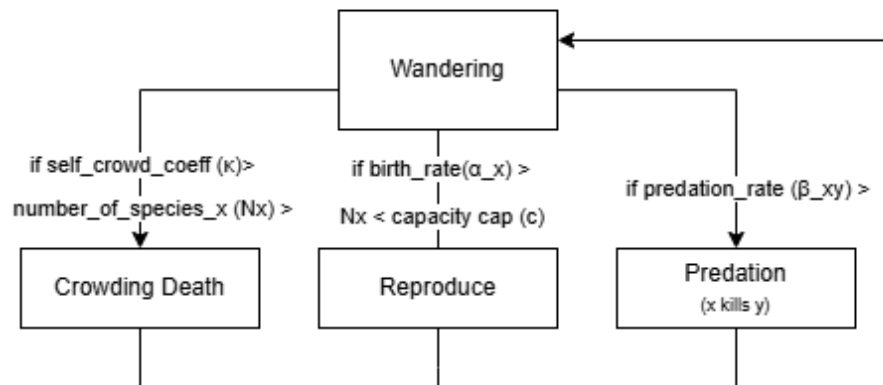
*Baseline Lotka-Volterra implementation*



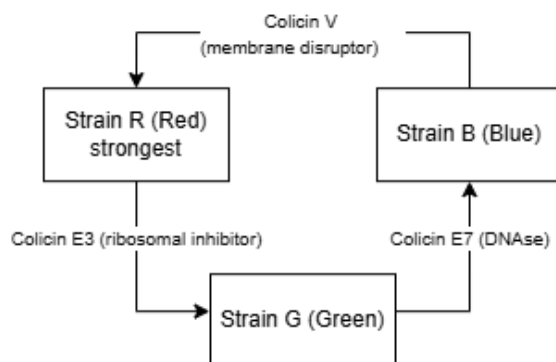
*Semi-realistic Lotka-Volterra implementation (prey and predator agents have different behavior flowcharts)*



*Rock-Paper-Scissors Lotka-Volterra implementation*



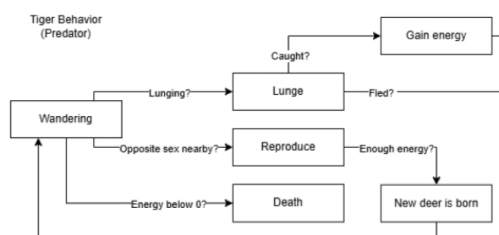
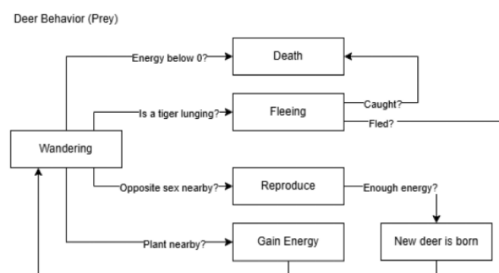
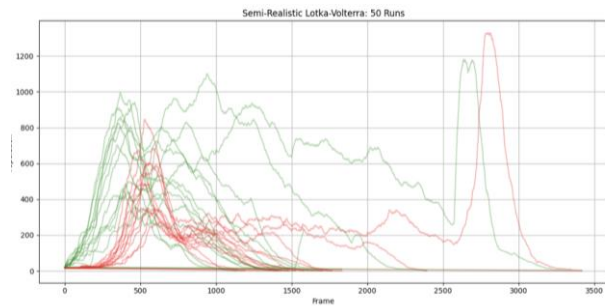
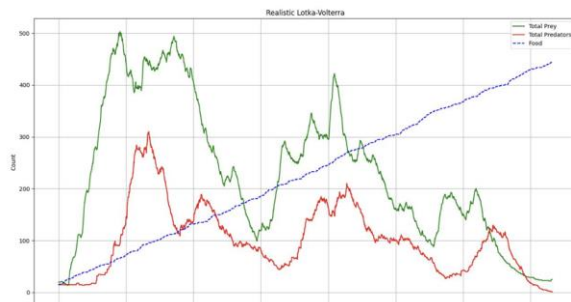
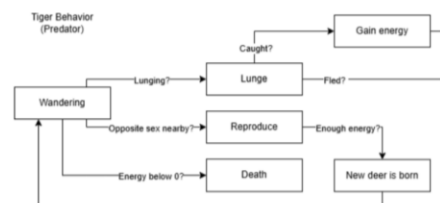
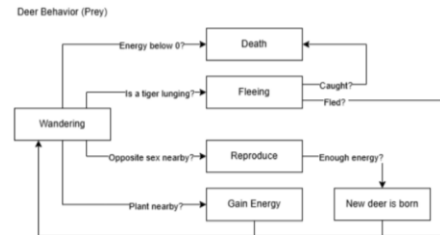
*Bacteria Lotka-Volterra implementation*



A *realistic* model was developed in the first week and presented, however due to it not being statistically testable (in terms of the way this study defined cyclic behavior and a stable system), it wasn't added to the second week's presentation (or the report). Below are some details about the *realistic* model:

## MODEL 3 – REALISTIC LVM

- The realistic model contains prey, predators, and food as agents with **state-based behavior** and with **energy, sex and age dynamics**. This model is fully **spatial-mechanics-based**, breaking the original baseline Lotka-Volterra assumptions. Prey have a prey-only site they can enter, where their speed gets lowered by 5% with a 95% chance (and gets sped up by 5% otherwise)
- Prey: consumes food to gain energy, reproduces if energetic and near an opposite-sex prey, based on a probability. Loses energy over time, and more energy if fleeing from a predator. Reproduction also costs energy, and to reproduce agents must be above a certain threshold. Dies if too old or out of energy.  
*Wandering* state on spawn and without predators nearby,  
*Flee* when predators are near,  
*Get Eaten* (over multiple frames) when caught by predators.
- Predator: Same reproduction, energy & age logic, consumes more if lunging towards a prey.  
*Hunting (Wandering)* state on spawn and without prey nearby,  
*Lunge* state when prey are near,  
*Eating* (over multiple frames) when a prey is caught.
- The eating duration is a shared parameter in the config, and both classes of agents spend the same amount of time/frames when a prey is caught and being eaten alive by the predator.
- Goal:** integrates classical LVM assumptions through behavioral rules rather than equations, preserving cyclic dynamics while introducing realism via spatial movement, energy thresholds, and proximity-based interactions.



[https://github.com/OlgaPiestrak/Collective\\_Intelligence\\_Project](https://github.com/OlgaPiestrak/Collective_Intelligence_Project)