Group Assignment: Natural Systems

COM3001 Simulation of Natural Systems
https://github.com/charalambosG-9/COM3001

N. Whyatt[a], C. Georgiades[a], L. Stephenson[b], E. Medcraft[b]

[a]*Department of Computer Science, University of Sheffield, Regent Court, 211 Portobello, Sheffield S1 4DP*
[b]*Department of Physics and Astronomy, University of Sheffield, Hicks Building, Hounsfield Road, Sheffield S3 7RH*

March, 2023

Abstract and Contributions

Natural systems are real world phenomena with complex underlying inter-correlated functions. To simulate a natural system is often desirable, as by understanding the underlying mathematical structures, predictions can be made as to the system's outcome. In this assignment, we demonstrate how natural systems can be simulated using mathematics using 1. The Lotka-Volterra equations, or predator-prey equations, an example of a first order system, through numerical methods and an agent-based model, and 2. Van der Pol's equations, an example of a dynamical second-order system, through numerical methods.

Throughout the project, we contributed equally to the individual tasks with all parties having a hand in the report writing and the code, presented in the linked git repository. Task 1 was completed by Nick W. and Charalambos G., and task 2 by Laura S. and Ethan M.

1. Task 1: Mathematical and Agent-based modelling

1.1. Literature Review

The predator-prey or Lotka-Volterra model, is a model used to simulate the dynamics of two interacting species: the prey and the predator. The model itself consists of two coupled differential equations, one describing the rate of change in the prey population and another describing the rate of change in the predator population.

The principles that would later be formalised in the predator-prey model were introduced by Alfred J. Lotka in 1925. In his book 'Elements of Physical Biology' [7], Lotka introduces the predator-prey model and provides an analytical solution for a simple version of the model. The general equations as they stand today were formulated in 1927 by Vito Volterra in his paper titled 'Variazioni e fluttuazioni del numero d'individui in specie animali conviventi' or 'Variations and fluctuations of the number of individuals in animal species living together' [15].

Although Lotka represented the predator prey model graphically in his original paper, M. L. Rosenzweig and R. H. Macarthur extended and refined his method in their paper published in 1963 [11]. They introduce a set of graphical techniques for representing these equations using the isoclines of the predator and prey populations to represent the points at which the rate of change of each population is zero, determining the behavior of the system over time. They use these methods to analyze the stability of predator-prey systems and derive conditions for stability. They show that stable systems must have certain relationships between the prey isocline, the predator isocline, and the initial population densities. They also demonstrate that these conditions can be used to predict the long-term behavior of the system, including the eventual equilibrium populations of the predator and prey.

A thorough description of the predator-prey model and its variations, along with other models of ecological interactions, is also given by Robert May in his paper written in 1974 [2]. May covers the traditional Lotka-Volterra model first, before moving on to alternative versions, such as the ratio-dependent model, which holds that the ratio of prey to predator densities affects the predator's functional response per capita. He also covers other ecological models, including parasitism, competition, and mutualism. He examines the equilibrium points of the predator-prey model and other ecological models, demonstrating how the stability of these equilibria depends on the model's parameter values. He concludes by discussing the concept of bifurcation, in which small changes in parameter values can lead to large changes in the dynamics of the system.

1.2. The Lotka-Volterra model

$$\begin{cases} \dfrac{dx(t)}{dt} = x(\alpha - \beta y) \\ \dfrac{dy(t)}{dt} = y(\delta - \gamma x) \end{cases}$$

Figure 1: The predator-prey model equations.

The variable x represents the number of prey while y represents the number of predators. The parameters α, β, γ and δ are positive real numbers describing the interaction between the two species. α represents the natural growth rate of the prey population in the absence of predation, β represents the predation rate, γ represents the natural death rate of the predator population in the absence of food and δ represents the rate at which the predator population grows when they consume the prey.

$$\begin{cases} \dfrac{dx(t)}{dt} = \alpha x - \beta xy \\ \dfrac{dy(t)}{dt} = \delta y - \gamma yx \end{cases}$$

Figure 2: The predator-prey model equations multiplied out.

The prey are assumed to have an unlimited food supply and to reproduce exponentially, unless subject to predation, which is represented in the first equation by the term αx. The rate of predation on the prey is assumed to be proportional to the rate at which the predators and the prey meet, which is represented by βxy. If either population is zero, there is no predation.

In the predator equation, δxy represents the growth of the predator population. The term γy represents the loss rate of the predators due to either natural death or emigration, and leads to an exponential decay in the absence of prey. Hence the equation expresses that the rate of change of the predator's population depends upon the rate at which it consumes prey, minus its intrinsic death rate.

1.3. Simulation using a numeric method

The predator-prey model was simulated using the Runge-Kutta 2 numeric method due to its stability compared to Euler's and Midpoint methods. This family of numerical methods approximates actual functions by mapping their progress at discretisation steps, gaining a degree of error that expands by a factor of the step size. Smaller discretisation steps lead to an increase in accuracy and a corresponding decrease in computational performance. However, by using different numerical methods, we can achieve less error.

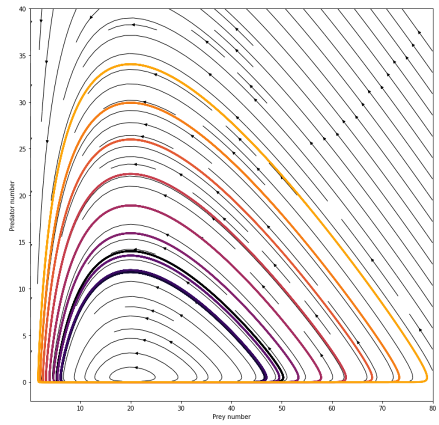

Figure 3: *A phase portrait analytically showing the result of the Runge-Kutta 2 method on the predator-prey model with varying starting conditions.*

The parameter values used to simulate the model were: $\alpha = 0.01$, $\beta = 0.02$, $\gamma = 0.4$, $\delta = 0.02$. For each subsequent trial, the starting number for the prey was increased by a certain amount while the starting number of predators was increased by a smaller amount. The resulting phase graph suggests that the populations of the two species exhibit a limit cycle behaviour. The triangular shape of the graph indicates that the two populations are following a cyclical pattern, with the predator population rising after the prey population reaches a certain threshold and then falling after consuming the majority of the prey. The round edges of the triangle suggest that the populations are not reaching a steady state but are instead oscillating in a periodic fashion. A larger limit cycle indicates that the populations are oscillating with greater amplitude which suggests that the model is more sensitive to parameter changes while a smaller limit cycle shows oscillation in a smaller amplitude and a more stable model.

1.4. Simulation using an agent-based model

An alternative way to model the prey-predator model is through *agent-based modelling*. These are stochastic models used to study the actions and interactions of autonomous agents on a shared plane and are useful for understanding system behaviour and the characteristics that govern outcomes within the model. Agents can range "from a 'software agent' or 'service/daemon', which might not behave very intelligently to an intelligent agent" where each agent is based on a model of artificially intelligent behaviour [9]. While there is no concrete definition of an agent, it is presented by [3] that they must be:

- **Autonomous**: agents are autonomous, information processing units that can potentially exchange this information with other agents and act/interact accordingly, to some extent.
- **Heterogeneous**: agents can be attributed differing attributes, and agents can be grouped into differing types and behaviours of similar autonomous individuals.
- **Active**: agents are active because they exert independent influence in a simulation.

These agent-based simulations are commonplace in the fields of life sciences and ecology, used to simulate the complex systems that make up our interconnected environment. Within this field, they are often shown under the broader heading "individual-based systems" [5]. In this task, we use Ecolab [4], a predator-prey modelling system using rabbits, foxes, and a controlling variable of 'grass', a resource used to ensure a finite population of rabbits, and hence a finite population of foxes.

A simple but effective graphical demonstration of the differences between the Lotka-Volterra equations and the agent - based model is the demonstration of the stochastic nature of the simulation. In Lotka-Volterra, your start position is a deterministic indicator of your future position - i.e., the number of prey and predators determines exactly the future course of the environment. Agent-based models are stochastic, and while two runs may have the same starting parameters, they may demonstrate radically different properties.

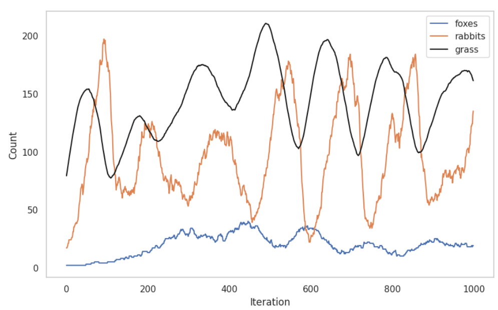

Figure 4: *1000 iterations of the Ecolab model, demonstrating an unstable start beginning to regulate.*

The two models in figures 4 and 5 were produced by the same parameters: 200 rabbits, 15 foxes, a starting 'grass' of 1, with a max 'grass' of 5 (meaning that the amount of food available will vary and will lead to fluctuations in the rabbit, and hence fox population) and attributes of 1 speed to rabbits, and 3 speed to foxes. However, they have radically different outcomes. In the first graph, enough rabbits survive the initial iteration to support enough foxes, which and with an abundance of food, rabbits spike, leading to a corresponding decrease in food and beginning wild oscillations that support a consistent fox population.

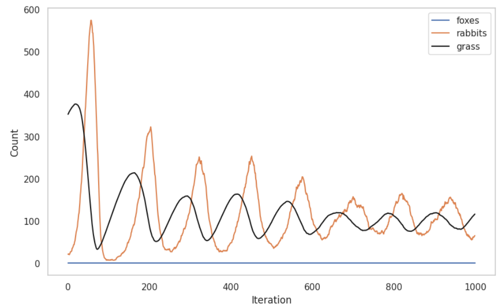

Figure 5: *1000 iterations of the Ecolab model, demonstrating foxes being unable to grow and the rabbit/grass relationship converging.*

However, the agent based system is subject to chance. Even if agent behaviour is deterministic - which it isn't entirely, as chances for failure create a change in the information of the whole environment and thus alter its course - starting positions and generations are often 'luck of the draw', and in positions with 15 foxes and 200 rabbits, often, the entire fox population does not survive past beyond first iteration, leaving a population of rabbits and grass to eventually converge on a relatively stable rabbit population.

Perhaps most importantly, although the above simulation describes a stable environment with no foxes and a stable amount of grass and rabbits, a freak incident in the agent model can occur, and a sudden spike in rabbits eating grass could lead to a spike in rabbits, a corresponding decrease in grass, and an effective extinction of both species. These events are not replicable to the same extent in Lotka-Volterra equations and the internal logics of agents with their stochastic success rates leads to a more sophisticated model of the environment, able to simulate responses of systems to freak occurrences over many generations.

1.5. Parameter Analysis

In this next section, we will maintain our above parameters, but run simulations varying the number of foxes. We chose this parameter after looking at the above (default parameters) result, where foxes more consistently than not were not able to initialise and form a stable population. A lower initial fox population may provide a higher probability to form a coherent fox population by being able to survive on a smaller population of rabbits in the early phase. It is interesting to know which starting conditions in an environment a fox could be introduced to, to maintain a healthy population of foxes and not drive them to immediate extinction. Therefore, our analysis will focus on modifying foxes to assess their impact on immediate fox extinction events within the model.

To analyse our population, we will modify the 'Nfoxes' parameter of the environment from the ranges 10 - 20, chosen due to their equidistance from the original parameter (15). Each one will only run for 100 (as opposed to 1000) iterations, as we are focusing our analysis on immediate fox extinctions in a given population. We will run each simulation five times, and count the number of immediate fox extinctions by counting the number of living foxes at the end of the simulation to determine if the number of foxes is the factor that leads to this event.

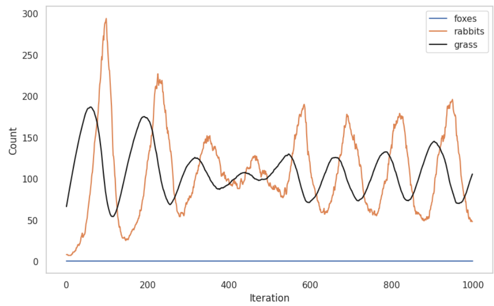

Figure 6: *1000 iterations of the Ecolab model, demonstrating foxes again being unable to grow and the rabbit/grass relationship converging.*

Our analysis initially left us more puzzled than before. While we had observed multiple immediate fox extinction events (attached in Figure 6, another one for good measure - these are replicable) our analysis of 5 iterations of 100 iteration runs across a range of foxes from 10 to 20 had produced zero extinction events, even on multiple runs of the program, and in fact demonstrated consistent fox populations slightly above the initial, as expected in a healthy growing population. It then occurred to us that the critical factor might in fact be the environment generation, as theoretically, if even the starting grass position was the same for all entities, despite all other positions being random, certain positions could tend more towards extinctions. Therefore, we regenerated the environment every run. Our results produced no extinction events. We produced a graph of our results.

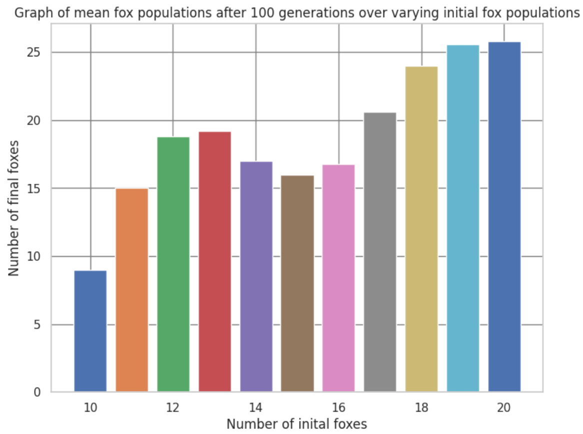

Figure 7: *100 iterations of differing starting fox populations and differing starting environments in search of an extinction event.*

While this somewhat anticlimactic result is disappointing, it points to a genuine need for further analysis to isolate the cause of extinction events replicable in graph form but not within the provided records functions. This is especially interesting because the graph isn't lying - without a fox presence, grass and rabbit populations tend towards convergence as they have done in all of the examples with an observed immediate extinction event. This suggests that the graph isn't at fault, but there is something more interesting at play that modulation of the fox variable doesn't bring out. A further analysis could perhaps be done by varying these foxes with a considerably smaller initial rabbit population, making the total threat of the fox population larger. Another alternative step could be to up the discretisation steps of analysis from 1 to something larger, like 5, and run more trials at more different intervals on the basis that a starting population of 10 and 11 are very similar.

2. Task 2: Dynamical Systems

2.1. Introduction

Van-Der-Pol's equation (1) is a second order differential equation which describes non-linear damping of oscillations. It describes systems which are non-Conservative, energy is added and dissipated by the system. Examples of systems described by the equation include a triode or RLC (resistance, inductor and capitor) electronic circuits as well as many biological, mechanical and acoustic systems [10]. The Van-Der-Pol exhibits interesting phenomena such as bifurcations, chaos and synchronisation which have been used to model the heartbeat for use in pacemakers [12].

$$\frac{d^2x}{dt^2} - \mu(1-x^2)\frac{dx}{dt} + x = 0 \tag{1}$$

The Van-Der-Pol is an example of the more generalised Liénard equation. The Liénard transformations can be applied to prove that the system has a limit cycle with the two dimensional form (2) [13].

$$\frac{dx}{dt} = y \tag{2}$$

$$\frac{dy}{dt} = \mu(1-x^2)y - x$$

The system approaches a limit cycle for small values of μ; when $|\mu| < 1$ energy is added to the system, when $|\mu| > 1$ energy is dissipated from the system [6]. The system tends to a stable limit cycle in phase space with large damping *(D'Acunto, 2006)*. When μ is very close to zero there is no damping and the system resembles a simple harmonic oscillator [14]. Figure 1 shows the Van-Der-Pol's solution for varying values of μ; It can be seen that for small values of the damping coefficient the solution resemble the harmonic oscillator. For large values of μ the solution tends to a relaxation oscillator[1].

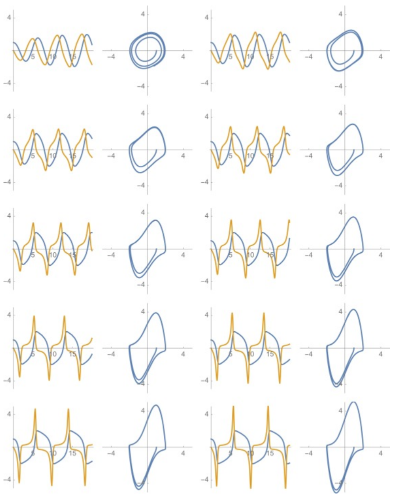

Figure 8: Solution to the Van-Der-Pol's equation for the initial condition $x(0) = 1, y(0) = 0$ has been shown with various values of μ. Figure has been reproduced from [1]

The forced Van-Der-Pol oscillator exhibits chaotic behaviour, forcing functions are given when the right hand side of equation 1 is changed from 0 to a function. Typically this function take a sinusoidal driving[8].

2.2. Task 2 Simulation Method

In order to simulate the Van-der-Pol oscillator, a numerical method has been chosen and implemented. For this simulation, Runge-Kutta (RK) methods were applied. This numerical method is a modification of Euler's method and has the general form shown in equation (3).

$$x(t + \delta t) = x(t) + \phi\delta t \tag{3}$$

Here, ϕ is called the effective slope which can be interpreted as a representative slope of the interval. Equation (4) shows the expression for ϕ.

$$\phi = a_1k_1 + a_2k_2 + ...a_nk_n \tag{4}$$

Different numbers of k terms can be considered, with a higher number of terms improving accuracy at the cost of computation. If only k_1 is considered, Euler's method is found. In this reports analysis, terms up to k_2 are considered i.e. RK2. Equations for k_1 and k_2 are shown in equations (5) and (6) respectively.

$$k_1 = f(x(t), t) \tag{5}$$

$$k_2 = f(x(t) + k_1\delta t, t) \tag{6}$$

For RK2, a_1 and a_2 are equal to 0.5. This numerical method has been implemented with code and applied to the characteristic equations of the Van der Pol oscillator.

2.3. Task 2 Simulation Results

Within the numerical methods simulation, a value of dt = 0.01 was chosen. The value of the constant μ was varied between 0 and 10 to show the effect on the solution and to see if it matches the theory discussed in the introduction. Figure 9 shows these solutions plotted on top of one another.

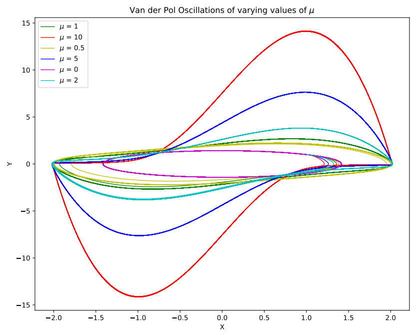

Figure 9: *A graph showing the solution to the Van der Pol oscillator dynamical system produced with Runge-Kutta 2 methods. Solutions with varying values of μ are shown in the same axis.*

It can be seen for lower values of μ that the solution for y is more circular as expected. For higher values, the solution shows a relaxation oscillator where damping comes into effect. The solutions converge well, with the solution for $\mu = 0$ being a very well defined circle. With the plots for each μ merged, the scaling of solutions can be seen and the starting point of (1,1) is visible as specified in the code.
The same solutions are plotted in Figure 10 but on separate axis to show them individually.

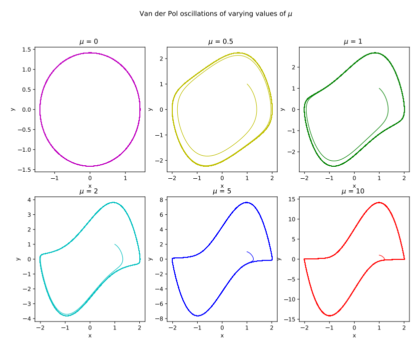

Figure 10: *A graph showing the solution to the Van der Pol oscillator dynamical system produced with Runge-Kutta 2 methods. Solutions with varying values of μ are shown separately to display how increasing μ shows a system tending towards a relaxation oscillator.*

Again the starting point of (1,1) can be seen for each solution and the shapes match expectations very well. The justifications for the choice of numerical method at dt value will now be discussed in the following section.

2.4. Discussion

The numerical method used to analyse the simulation was RK2. This method is a modification of Euler's method. It was chosen because it is more accurate then Euler's method and the midpoint numerical methods. This is because RK2 has local error of order δt^3 compared to Euler's larger δt^2. RK2 has the accuracy of a Taylor series expansion without the requirements of calculating higher order derivatives which takes more numerical analysis. More terms of RK can be used for even higher accuracy but only 2 terms were used as this is one extra term of information compared to Euler's. By doing this, Euler's main flaw is overcome. This flaw is that Euler's method assumes the derivative at the start of a given interval can be applied to the rest of the entire interval.

It is also worth noting that if a_1 is set equal to 1 and a_2 is 0, the midpoint method is formed from RK 2. This again shows how RK is an extension of other numerical methods and thus is the golden standard in numerical methods.

The reasoning behind the selection of dt will now be discussed. Figure 11 depicts solutions to the Van-Der-Pol's equation with varying discretisation steps in the range 0.02-0.05 s. As the time interval increases there is a larger uncertainty for a given x value shown by the increased line width, the opposite is true for smaller discretisation steps where the solution converges. A value of dt=0.01 shows the most accuracy.

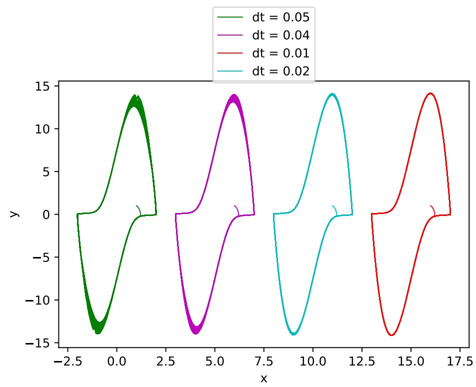

Figure 11: *A graph showing the solution to the Van-Der-Pol oscillator dynamical system produced with Rugne-Kutta 2 methods. Solutions with varying values of dt are shown but with a fixed value of μ. This is to demonstrate how lowering dt means the solutions is less uncertain.*

However, their exists a trade-off between the time interval and the computational time for the solution to be generated. Figure 12 demonstrates the relationship between computational time and the discretisation step dt. It was not possible to run solutions for smaller time intervals as the computational time costs were too great, while larger time intervals lead to significant errors in the solution that diverge from expected shapes.

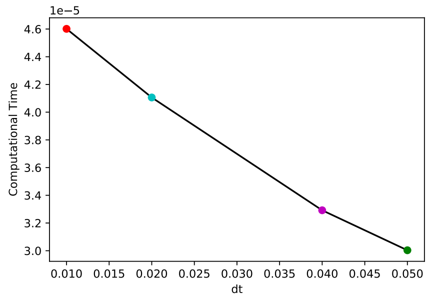

Figure 12: *Computational time in seconds as a function of time interval for generating the solutions of the Van-Der-Pol equations using the RK2 numerical method. Each time interval has been assigned the same colour as Figure 11*

The relationship in Figure 12 appears to be linear. A value of dt=0.01 gave best accuracy with reasonable computational time for the purpose of this report.

References

[1] M. L. Abell and J. P. Braselton, «Chapter 7 - applications of systems of ordinary differential equations», in *Differential equations with mathematica (fifth edition)*, edited by M. L. Abell and J. P. Braselton (Academic Press, Jan. 1, 2023), pp. 385–421, ISBN: 978-0-12-824160-8, 10.1016/B978-0-12-824160-8.00012-7.

[2] R. Arditi and L. R. Ginzburg, «Coupling in predator-prey dynamics: ratio-dependence», Journal of Theoretical Biology **139**, 311–326, ISSN: 0022-5193 (1989) https://doi.org/10.1016/S0022-5193(89)80211-5.

[3] A. Crooks and A. Heppenstall, «Introduction to agent-based modelling», in (Jan. 2012), pp. 85–105, ISBN: 978-90-481-8926-7, 10.1007/978-90-481-8927-4_5.

[4] M. Croucher, *Ecolab: a high performance agent-based modelling system for c++*, 2017.

[5] V. Grimm and S. F. Railsback, *Individual-based modeling and ecology* (Princeton University Press, 2013).

[6] I. Kovacic and R. E. Mickens, «A generalized van der pol type oscillator: investigation of the properties of its limit cycle», Mathematical and Computer Modelling **55**, 645–653, ISSN: 08957177 (2012) 10.1016/j.mcm.2011.08.038.

[7] A. J. Lotka, *Elements of physical biology*, Vol. 21, 82 (Temporary Publisher, 1926), pp. 341–343.

[8] J. M. Mahaffy, «Math 537 - ordinary differential equations - lecture notes – method of averaging», Lecture Notes, Lecture Notes, Department of Mathematics and Statistics Dynamical Systems Group Computational Sciences Research Center San Diego State University, 2019.

[9] M. A. Niazi and A. Hussain, «Agent-based computing from multi-agent systems to agent-based models: a visual survey», CoRR **abs/1708.05872** (2017).

[10] C. M. Pinto and J. T. Machado, «Complex-order forced van der pol oscillator», Journal of Vibration and Control **18**, 2201–2209, ISSN: 1077-5463, 1741-2986 (2012) 10.1177/1077546311429150.

[11] M. L. Rosenzweig and R. H. MacArthur, «Graphical representation and stability conditions of predator-prey interactions», The American Naturalist **97**, 209–223 (1963) 10.1086/282272.

[12] A. M. dos Santos, S. R. Lopes, and R. L. R. L. Viana, «Rhythm synchronization and chaotic modulation of coupled van der pol oscillators in a model for the heartbeat», Physica A: Statistical Mechanics and its Applications **338**, 335–355, ISSN: 0378-4371 (2004) 10.1016/j.physa.2004.02.058.

[13] J. Sugie and K. Ishibashi, «Limit cycles of a class of liénard systems derived from state-dependent impulses», Nonlinear Analysis: Hybrid Systems **45**, 101188, ISSN: 1751-570X (2022) 10.1016/j.nahs.2022.101188.

[14] *Van der pol equation - encyclopedia of mathematics*, https://encyclopediaofmath.org/index.php?title=Van_der_Pol_equation (visited on 03/27/2023).

[15] V. Volterra, *Variazioni e fluttuazioni del numero d'individui in specie animali conviventi*, Vol. 2 (JSTOR, 1927), pp. 31–113.