On Lotka-Volterra's predation equations

Valérie Maleappa, Yann Bellardi, Jules Guerault

Directed by :
Adrien Laurent

1 Introduction

1.1 Historical origins

In 1925, Italian marine biologist Umberto d'Ancona noticed a surprising pattern followed by fish stocks in the Mediterranean Sea: different fish populations periodically grew and shrank, in a seemingly connected way. To better understand these dynamics, d'Ancona asked the mathematician Vito Volterra to figure out a mathematical model for it, which he did in his 1926 paper *Variazioni e fluttuazioni del numero d'individui in specie animali conviventi*. This is how the predator-prey model was first established, later put in equivalence with Alfred Lotka's chemical equations. Let us try to understand how this model was built.

1.2 Naive Approach

First of all, let us try to find an explicit way to predict these dynamics. We want a function expressing both populations for any given time. Therefore, the equation will be in the form $x = f(t)$ and $y = g(t)$. As we want to introduce interactions between the species, each population will depend on each other. Thus we will have $x = f(t, y)$ and $y = g(t, x)$. However, something is troubling : to express the population of preys, we need to know the predator's, and vice versa. In some way, our approach asks us to know the prey's population before calculating it. Because of that, we will have to build a model comporting differential equations to express these populations : an equation that expresses a relationship between functions (the population) and their derivatives (the population's changes).

1.3 Model construction

First things first, let's assume that when they are not hunted by predators, the prey's population has a tendency to grow, and that this growth is proportional to the population's size because the more they are, the more they breed. Mathematically, that means that the population's derivative is proportional to its size. Thus we can assume that the function describing the prey's population will be a solution of an equation of the form :

$$x' = ax$$

On the contrary, the predators' population has a tendency to dwindle, and this dwindling is equally proportional to the size of the population because the more they are, the more the lack of food will be felt. We can then establish a temporary equation for the predators' population which will be of the form :

$$y' = -cy$$

Now, let's introduce interactions between species. The prey's population, initially growing, will in time know a shrink as the predators' population grows. Thus, we can consider that the growth rate is not constant and diminishes proportionally to the number of predators. Then the equation becomes:

$$x' = (a - by)x$$

Vice versa, the predators' population, initially dwindling, will increase when enough prey are available to hunt. Their growth rate will augment exponentially proportional to the number of prey. We finally have the equation

$$y' = (-c + dx)y$$

Thus, population functions will be the solutions to the following problem :

$$\begin{cases} x' = (a - by)x \\ y' = (dx - c)y \end{cases} \tag{LV}$$

1.4 Questions and Problems

At first glance, we could be tempted to say that we simply need to resolve these equations to find the expression of the population. However, these differential equations are not linear, and so explicit resolution methods cannot be applied. In general, there is no explicit solution to the equation. It will then be a matter of studying the different tools at our disposal. Firstly, to prove the existence of solutions. Then, to seek methods to find values for these solutions. With that in mind, our research question will be :

How to approximate solutions for the Lotka-Volterra's equation system ?

Contents

2 Properties of hamiltonian systems

Proposition 2.1. *Let $Y \in \mathbb{R}^{2d}$ and $J = \begin{pmatrix} 0 & Id \\ -Id & 0 \end{pmatrix}$. A Hamiltonian system is a system that can be written as $Y' = J\nabla H$, with $H \in C^2(\mathbb{R}^{2d} \to \mathbb{R})$*

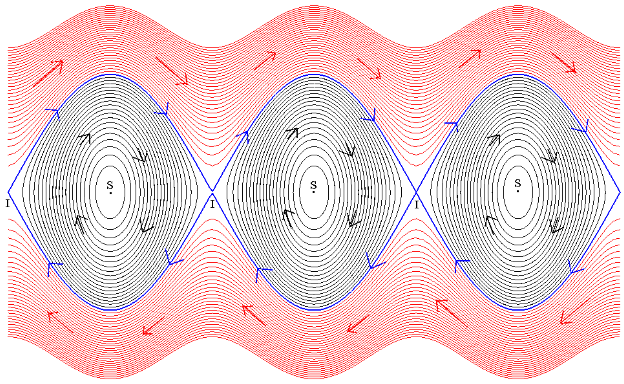


Figure 1: Phase space of the pendulum

A classic example of such a system is the following: $\begin{matrix} q' = p \\ p' = -q \end{matrix}$ Then, with $Y = \binom{p}{q}$ we can see that $Y' = \binom{-q}{p} = JY$. Therefore, with $H = \frac{||Y||^2}{2}$ we have $\nabla H = Y$, thus $Y' = J\nabla H$

This system is called the pendulum because p and q describe respectively the position and momentum of a pendulum.

2.1 Phase space

The phase space of a Hamiltonian system is the set of all possible states taken by the system. In the case of the pendulum, the phase space is a 2d plane whose coordinates are the position and momentum of the pendulum at any given time. Since the system equations allow us to predict the evolution of the system from any given state, we can picture a vector field on the phase space of the system giving the predicted evolution at any given time, as seen in figure 1. This vector field is called the flow of the equation.

2.2 Finding solutions

2.2.1 Local solution

Let us consider the following Cauchy problem :

$$\begin{cases} u'(t) = f(u(t)) \\ u(0) = \quad u_0 \end{cases} \tag{CL}$$

with $f : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ a vector field and $u_0 \in \Omega$ and $\Omega \subset \mathbb{R}^n$

To prove the local existence of solutions, we use the Cauchy-Lipschitz theorem :

Theorem 2.2 (Cauchy-Lipschitz)**.** *if f is locally Lipschitzian, then there exists a unique maximal solution $u \in \mathcal{C}([0, T[, \mathbb{R}^n)$ de (CL) pour $t < T$*

Note that since H is of class $\mathcal{C}^2$, ∇H is of class $\mathcal{C}^1$ and therefore locally Lipschitzian, and that this theorem leads to the following corollary:

Corollary 2.3. *For two different initial conditions, the solutions do not intersect.*

2.2.2 Global existence, hamiltonian

Within the scope of application of the Cauchy-Lipschitz theorem (that is f of CL locally Lipschitzian), the theorem below gives a sufficient condition to prove global existence.

Theorem 2.4 (compacts break)**.** *Let u be the maximal solution of CL defined for $t < T$. If u is bounded on $[0, T[$, then $T = +\infty$*

Further precisions can be found in [3]

2.3 Properties of solutions

The following theorems are fundamental properties of the solutions of Hamiltonian systems.

Theorem 2.5. *For every solution Y(t) of the hamiltonian system $Y'(t) = J\nabla H(Y(t))$, $t \mapsto H(Y(t))$ is constant, in other words $H'(Y(t)) = 0$*

Proof. $H'(Y(t)) = \nabla H(Y(t)) \cdot Y'(t)$ By definition, $Y'(t) = J\nabla H(Y(t))$. Therefore, $H'(Y(t)) = \nabla H(Y(t)) \cdot J\nabla H(Y(t)) = 0$ because $\nabla H(Y(t))$ and $J\nabla H(Y(t))$ are orthogonal. ☐

In the case of the pendulum, $H(Y(t)) = \frac{p(t)^2+q(t)^2}{2}$ therefore, $H'(Y(t)) = p'(t)p(t) + q'(t)q(t) = q(t)p(t) - q(t)p(t) = 0$. Let f be a linear map in $\mathbb{R}^{2d}$. Therefore there is a matrix $A \in M_{2d}(\mathbb{R})$ that allows $f(X) = AX$. We say that the flow described by f is symplectic if $AJA^T = J$. This definition can be extended to non-linear maps using the Poisson brackets.

Theorem 2.6 (Poincaré theorem)**.** *The flow of any Hamiltonian system is a symplectic transformation.*

Theorem 2.7 (Liouville theorem)**.** *The volume of any part of the phase space is preserved by the evolution of the system.*

The proof of both these theorems can be deduced from the first one. However, the exact proof is out of the scope of the current paper.

3 Application to LV

3.0.1 Cauchy

Let us apply the methods presented in 2.2 to the (LV) system. Let u be equal to $\begin{pmatrix} x \\ y \end{pmatrix}$, to have a form close to (CL).

$$\begin{cases} u'(t) = f(u(t)) \\ u(0) = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \end{cases} \quad avec\ f = \begin{pmatrix} x(a - by) \\ y(-c + dx) \end{pmatrix}$$

The function f is derivable and continuous on the interval $[0, T)$, it is therefore of class $\mathcal{C}^1$ on the same interval. Thus, we can ensure the local existence and uniqueness by application of the Cauchy-Lipschitz theorem. We can then prove that all solutions are positive if $x_0 \geqslant 0$:

Proposition 3.1. *If $x_0 \geqslant 0$, then $x(t) \geqslant 0$.*

We have $x_0 \geqslant 0$. If $x_0 = 0$, then for all $t < T, x(t) = 0$. Furthermore, solutions do not intersect. We thus have $x(t) > 0$ if $x_0 > 0$. The same goes for y.

We can notice these results in figure 2, a phase portrait of the (LV) equations. A phase portrait is the representation of the trajectories of a system for different initial conditions. As the solutions are two variables depending on t, we can define a parametrized curve that represents on a coordinate plan $(x(t), y(t))$ for $t \in [0, +\infty[$. For this purpose, the figure has $x(t)$ as the abscissa and $y(t)$ as the intercept, instead of t as the abscissa. The solutions are periodical, therefore the curves will be closed. On this figure, each of the trajectories represents a different initial condition, which allows us to appreciate the fact that trajectories seem distinct and strictly positive. This is coherent with the previous results.

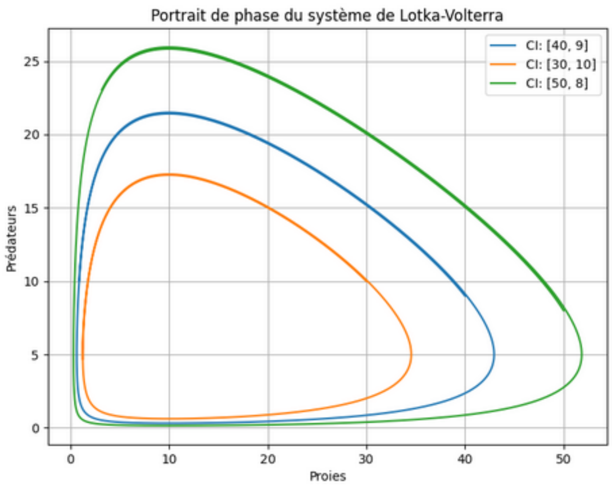


Figure 2

3.0.2 Existence of global solutions and Hamiltonian operator

Proposition 3.2. *Let $H : \mathbb{R}^2 \to \mathbb{R}$ be defined for $x, y > 0$ by*

$$H(x, y) = dx - c\ln x + by - a\ln y$$

Then H is an integral first for the (LV) system, that is if $(x(t), y(t))$ is a solution of (LV) on $[0, T)$, then

$$\forall t < T, \quad H(x(t), y(t)) = cste$$

Proof. We check that $\frac{\partial}{\partial t} H(x(t), y(t)) = 0$:

$$\frac{\partial}{\partial t} H(x(t), y(t)) = dx(t)' - \frac{cx(t)'}{x} + by(t)' - \frac{ay(t)'}{y}$$

We replace $x(t)'$ and $y(t)'$:

$$\frac{\partial}{\partial t} H(x(t), y(t)) = d(a - by(t))x(t) - \frac{c(a - by(t))x(t)}{x(t)} + b(dx(t) - c)y(t) - \frac{a(dx(t) - c)y(t)}{y(t)}$$

We simplify and develop :

$$\frac{\partial}{\partial t} H(x(t), y(t)) = adx(t) - bdy(t)x(t) - ac + bcy(t) + bdy(t)x(t) - bcy(t) - adx(t) + ac = 0$$

We thus find $\frac{\partial}{\partial t} H(x(t), y(t)) = 0$, and we deduce that $H(x(t), y(t)) = cst$. $\square$

Thanks to this first integer, we can prove the following lemma :

Lemma 3.3. *The maximal solution* $(x(t), y(t))$ *is bounded.*

Proof. We assume to simplify that $x_0 > 0$ and $y_0 > 0$, because x et y are strictly positive since they describe animal populations, and if $x = 0$ or $y = 0$, the solution becomes evident (and thus bounded : if $x_0 = 0$ or $y_0 = 0$, the system becomes a 1st-order differential equation).

Let x be such that $\lim_{t\to+\infty} x(t) = +\infty$. In that case, $\lim_{x\to+\infty} H(x(t), y(t)) = +\infty$: by comparative growth, when $\lim_{x\to+\infty}$, $dx(t) - c\,ln\,x = dx(t)$. However, $H(x(t), y(t)) = cte$; and thus this result is absurd. Furthermore, $x(t)$ is in that case strictly superior to 0. $x(t)$ is thus bounded and $x \in]0, +\infty[$. We can prove that y is bounded in the same way. □

4 Numerical Integration

4.1 Numerical Methods

We are trying to approach the solutions of the (LV) system. There are many ways to do that, and the choice depends on the accuracy wanted or the preservation of geometric properties. A first method that we have implemented is explicit Euler.

4.1.1 Order and numerical integrator

Definition:
We call a numerical integrator with 1 step a mapping with real values :

$$U_{n+1} = \phi_h(U_n)$$

with $\phi_h : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$
and $U_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$.

We remind the vector field associated to the (LV) system is:

$f : \mathbb{R}^2 \to \mathbb{R}^2$, with $f(x, y) = \begin{pmatrix} x(a - by) \\ y(-c + dx) \end{pmatrix}$.

4.2 Order of a numerical method

Let y be the exact solution. We define the local and global order as follows:
Local error in order p : $|y(1) - \phi_h(h)| \leqslant C * h^{p+1}$
Global error in order p: $|y(k) - \phi_h(tk)| \leqslant C * h^p$, for all integer k with $0 \leqslant k \leqslant N$.

4.2.1 Explicit Euler

Explicit Euler method can be written as:

$$U_{n+1} = U_n + hf(U_n). \tag{4.1}$$

Here, $\phi_h = (Id + hf)$

4.2.2 Implicit Euler

By taking the previous notations, we do a slightly different estimate:

$$U_{n+1} = U_n + hf(U_{n+1}) \tag{4.2}$$

This method requires the use of the fixed point method in the implementation (for each step, we have to resolve an equation).

However, we observe quite quickly in the numerical simulations on Python that both methods do not preserve the Hamiltonian nor the geometric properties.

Thus, we consider a new method, inexpensive and with surprising results: Symplectic Euler.

4.2.3 Symplectic Euler

Let's remember that our sequence U_n contains information on the preys' and the predator populations. We have $U_n = (x_n, y_n)$. Unlike the previous methods, we process x_n and y_n differently.

$$x_{n+1} = x_n + hf(x_n, y_n) \tag{4.3}$$

$$x_{n+1} = x_n + hf(x_{n+1}, y_n) \tag{4.4}$$

The only difference with the explicit method is in the argument of the f function for y_{n+1}. We take x_{n+1} and not x_n. This method is order 1, but what is particularly interesting is that it preserves very well the Hamiltonian! The error is exponentially tiny, the implementation is simple without any supplementary equation to resolve (unlike implicit Euler's method), and yet it preserves well the geometric properties.

4.3 Cranck-Nicolson

Another symplectic method, and that requires the resolution of an equation for each step (in our case, we use the fixed point method) is the Cranck-Nicolson's method, of order 2:

$$U_{n+1} = U_n + \frac{h}{2}(f(U_n) + f(U_{n+1})) \tag{4.5}$$

If we now want methods with a Highter order, we use Runge-Kutta's method of order 4.

4.4 Runge-Kutta's method of order 4

A very accurate method is the 4th order Runge-Kutta method, RK4 :

$$U_{n+1} = U_n + \frac{h}{6}(f(k_1) + f(k_2) + f(k_3)) + f(k_4)) \tag{4.6}$$

with

k$_1 = U_n$;
k$_2 = U_n + \frac{h}{2}f(k_1)$;
k$_3 = U_n + \frac{h}{2}f(k_2)$;
k$_4 = U_n + hf(k_3))$.

This method is order 4. However, it does not preserve the Hamiltonian, and in long time, the geometric properties are not preserved. Thus, the choice of the method depends on which properties we want to find in our numerical solutions.

4.5 Efficiency

The efficiency of these methods is not equal : a symplectic method preserves better the Hamiltonian and the geometric properties of the flow seen in 2.3. RK4 - order 4, is more accurate in a short time but does not preserve the Hamiltonian in a longer time.

4.6 Symplectic Methods

Our system is Hamiltonian by a change of variable :

$$p(t) = ln[\frac{c}{d}x(t)],\ q(t) = ln[\frac{a}{b}y(t)] \tag{4.7}$$

With x and y the prey and predator's populations, always positive.

$$p'(t) = a(1 - e^q),\ q'(t) = -c(1 - e^q). \tag{4.8}$$

And the Hamiltonian G verifies : $G(p, q) = a(q - e^q) + c(p - e^p)$. Thus, $\begin{pmatrix} p' \\ y' \end{pmatrix} = J\nabla G$.

Proposition:

Symplectic Euler is symplectic.
For the pendulum, we can see the proof in the section VI.3 of the Geometric Numerical Integration.

For the (LV) system, we can't express the link between $H(y_{n+1})$ and $H(y_n)$ easily by calculus. Unfortunately, caught up by time, and due to the limited size of the report, we can't make the entirety of the proof for the (LV) system by hand, but here are some tools and main reasoning: [2].

We try to make a Taylor development of the numerical solution. We have to learn some tools of derivatives for functions of multiple variables (Leibniz' rule, linearity), and then use B-series, by learning more about Butcher trees. [1] Then, we have to make Backward Error Analysis to find a close Hamiltonian $\tilde{H} = H + O(h^)$.

Our system is a Hamiltonian as seen in 2 by the transformation seen above, this close Hamiltonian is preserved by Euler's symplectic method. We could also use Poisson Structure shown in Geometric Numerical Integration, to find a close Hamiltonian by observing that $f = \begin{pmatrix} 0 & xy \\ -xy & 0 \end{pmatrix} J\nabla H$. This close Hamiltonian is exactly preserved by Euler's symplectic method. We then have that the error is tiny in exponentially long time.

5 Numerical results on Python

5.1 Code's constructioon

5.2 Results in Python

This section is dedicated to our results in Python for explicit Euler, symplectic Euler, and RK4 methods.

```python
def euler(F, t0, tf, y0, n): #y0 est une liste de 2 éléments : nb initial de lapins, et de renards.
    h = (tf - t0) / n
    t = 0
    Y = [y0]
    T = []
    for k in range(n):  # Euler explicite
        Y.append([Y[k][0]+F(Y[k],t)[0]*h,Y[k][1]+F(Y[k],t)[1]*h])
        t = t + h
        T.append(t)
    return T, Y # on est sur des array des valeurs, y a 2 var.
# X=(x,y), X'=(x',y')
a = 2
b = 3
c = 2
d = 1
def F (y, t): #F est la fn représentant l'opérateur dérivée du système LV
    U, V = y[0], y[1]
    return [U * (a - b * V), V * (d * U - c)]
```

Figure 3: Caption

We take t_0=0, $t_f = 50$, $y_0 = [1, 2]$, and 10 000 points. The prey population is represented in orange and the predator one in blue. (a,b,c,d)=(2,3,2,1).

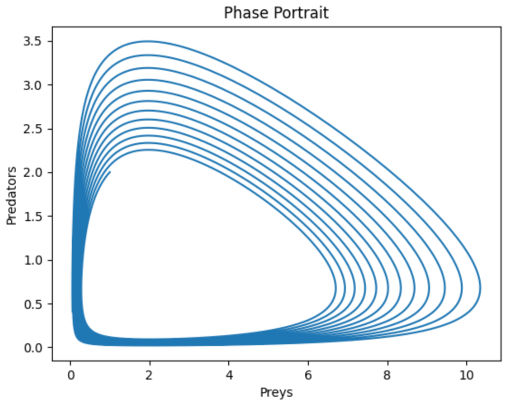
It allows us to observe the results of methods seen previously. The Hamiltonian is well preserved for Symplectic Euler's method (figure 5), and we can see the geometric properties (periodicity). For Explicit Euler's method (figure 4), the Hamiltonian explodes and the populations grow very fast. For RK4's method (figure 6), it is hard to make a conclusion when applying the method for short time periods, even though we can see that the Hamiltonian does not seem to be preserved.

We also have the phase protract for explicit Euler and Symplectic Euler, we see Symplectic Euler is more efficient since it explods with Explicit Euler, in figure **??**

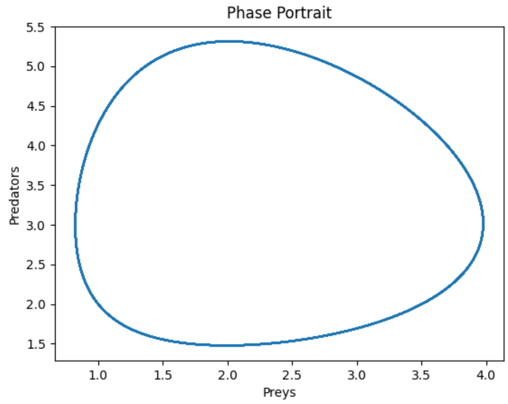
6 Conclusion

For our system, no explicit methods are able to give the exact values for the solutions. Nonetheless, some numerical methods can give accurate approximations of the solutions' values on a limited time period.

The choice of the methods depends on the properties the user is willing to preserve : if the goal is to minimize the absolute error on a short period of time, high order methods, such as 4th-order Runge-Kunta would be best advised. However, if the choice is to preserve geometric properties of the solutions such as the theorems referenced in 2.3, it is necessary to use a symplectic method instead, such as the symplectic Euler method or Cranck-Nicolson. Furthermore, it is important to note that the



(a) Phase Portrait of Explicit Euler



(b) Phase Portrait of Symplectic Euler

Phase Portrait comparison - Explicit and Symplectic Methods

higher order methods tends to be highly expensive in process power and therefore hard to implement in informatical simulations on long periods of time. Thus, it is the user's choices to decide how precise they want the approximation to be, given the resources they have at hand.

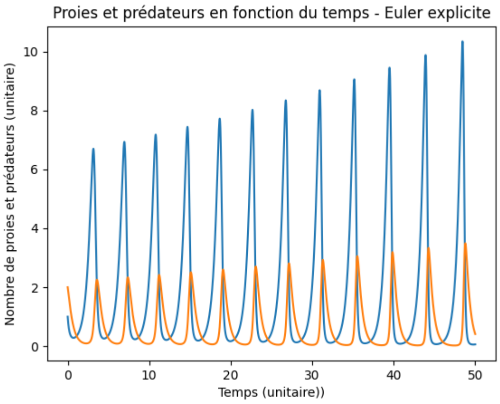
The introduced methods, especialy Symplectic Euler, Cranck-Nicolson, RK4, provide a good accuracy/efficiency ratio for our purpose.

References

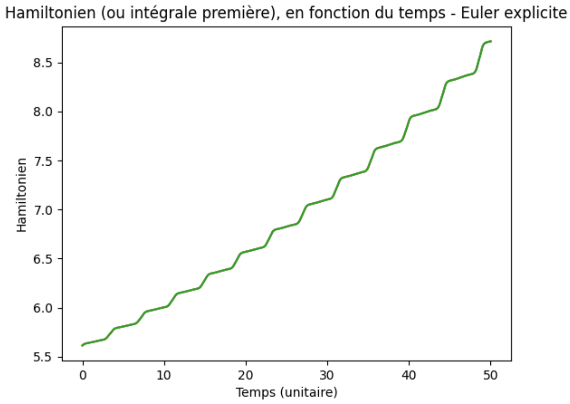
[1] John C. Butcher. *B-series and Algebraic Analysis*, pages 99–149. Springer International Publishing, Cham, 2021.

[2] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006. Structure-preserving algorithms for ordinary differential equations.

[3] Gregory Vial. Le système proie-prédateur de volterra-lotka. year : 2018.

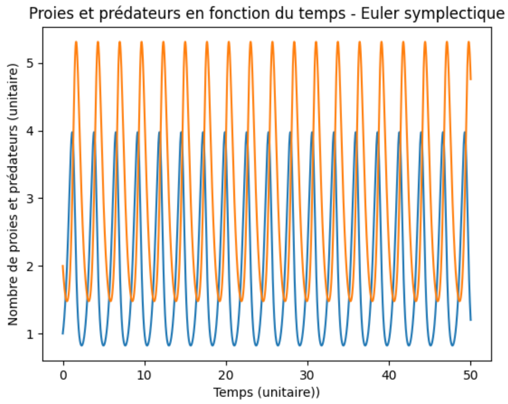


(a) Populations

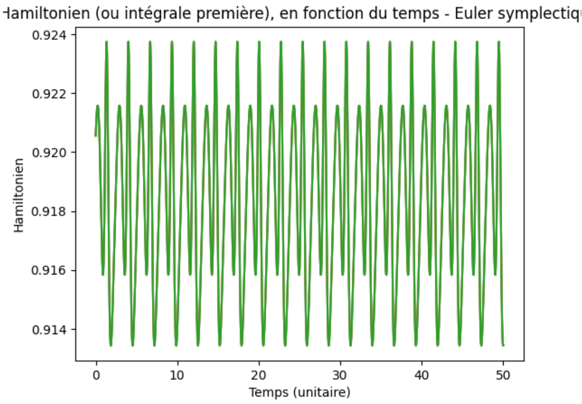


(b) Hamiltonian

Figure 4 : Explicit Euler

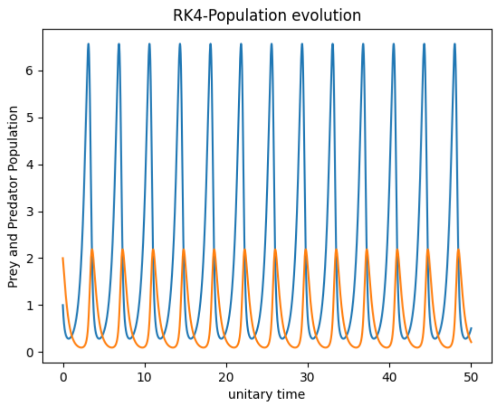


(c) Populations

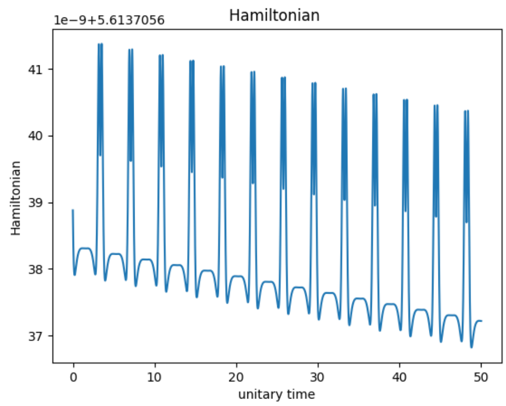


(d) Hamiltonian

Figure 5 : Symplectic euler



(e) Populations



(f) Hamiltonian

Figure 6: 4th order Runge-Kutta

Figure 5: Populations and Hamiltonians as approximated by various numerical methods