Computational Lab Report
Graphical Methods

Report by: **Jamie Somers**

January 5, 2022

Abstract

Python is extremely powerful programming language for physicists and mathematicians, not just for its ability to carry out complex maths equations but also to be able to graph representations for these mathematical equation. We are going to be using quiver plots to graphically represent the potential solutions to differential equations. In this report we will be including plots containing vector fields with directions and magnitudes as well as contour plots which gives a background color representing the magnitude. These plots represent real physical systems specifically population, Lotka-Volterra model and limit cycles. We do this to explore the different graphical methods of displaying differential equations and when they don't accurately represent the real world models they are trying to reproduce.

1 Introduction

This report will be building on what we have previously done regarding Euler's method and more importantly the Runge-Kutta fourth order method of solving differential equations in python, however we will be moving away from these methods and focusing on quiver plots which are used to produce vector fields. These quiver plots will graphically describe the solution to differential equations at many different points this is extremely useful for representing different types of differential equations. In this lab report we will graphically be describing population simulations from differential equations. The two population models are known as the Verhulst model and the Lotka-Volterra model respectively. In this report we will be commenting on the difference between two graphical models as well as how these graphical differences represent the theoretical differences inherent in the two models. We will then discuss some more mathematically abstract models which we believe will help us learn more about graphical methods in Python.

2 Background and Theory

The first thing we cover in this report is the theory surrounding the quiver plots produced in Python. Vector fields describe the direction and magnitude of vectors at a number of points in a plane, vector fields are common in electromagnetism and also in a field of mathematics known as vector calculus. Its no surprise then that Pythons plotting package known as matplotlib comes with the ability to create graphical representations of vector fields included in the package.

Throughtout the report we will be making reference to vectors, vectors are any object which has both a direction as well as a magnitude. We often display vectors graphically as arrows where the head of the arrow denotes the direction and the length of the arrow represents the magnitude.

Taking the vector field example one step further we can create a certain type of plot known as a stream plot, this stream plot creates trajectories or field lines from vector fields. Each of the stream lines represents a possible solution to a differential equation, this way we can visualise solutions to differential equations without actually solving them.

In this experiment we will be using quiver plots and stream plots to show the expected behaviour of multiple differential equations, this is because quiver plots and by extension stream plots are pythons way of graphing vector fields, quiver plots are extremely useful because we can change some of the initial conditions to change how the vector field behaves, for example one of these conditions is the number of vectors that appear per grid point this allows us to increase or decrease the density of vectors.

The first part of this experiment concerns itself with getting familiar with these initial conditions before returning to the harmonic oscillator and damped harmonic oscillator equations we have worked with previously.

The second part of this experiment concerns itself with graphing vector fields showing the potential solutions for population differential equations. One such equation is the logistic equation known as the Verhulst model for population. The equation is as follows:

$$\frac{dx}{dt} \equiv \dot{x} = rx(1 - \frac{x}{K}) \tag{1}$$

Where K is the carrying capacity or the maximum population of the system, r is the rate of growth or decay of the system and $x(t)$ represents the population at a certain point in time.

This equation is considered to be a simple model for how population reaches equilibrium and only focuses on one population and doesn't account for how one population may have an impact on another population such as a hunter / prey model which we will explore in more detail.

We will also explore the Lotka-Volterra (LV) Model which models two populations simultaneously and how the two populations impact each other using rabbits and foxes as the example. The equations for the Lotka-Volterra (LV) Model is as follows:

$$\frac{dx}{dt} = ax - bxy \tag{2}$$

$$\frac{dy}{dt} = cxy - dy \tag{3}$$

Where $x(t)$ is the population growth of the rabbits, $y(t)$ is the population growth of the foxes, a defines the growth rate, b is a parameter that measure how effective the foxes are at hunting rabbits, c is a parameter that defines how beneficial rabbits are to the foxes and d is a measure of this fox death rate.

We can see that this model uses two differential equations to represent each population and the population of one species has an impact on the population of the other, more specifically a higher population of foxes reduces the population of the rabbits and the decrease in rabit populations results in a decreased fox population. When we analyse the graphs obtained from this model we will discuss the model in more detail and its strengths and weaknesses as a biological model to represent population.

This is all the theory we need to begin analysing the data collected from the python programmes in the next section of the report. We will not go into the results collected during the experiment.

3 Results

Week 9

Determining fixed points			
det(J)(Δ)	Tr(J)(τ)	τ^2-4Δ	Type of fixed point
-6 $<$ 0	$1 > 0$	$1^2 - 4(-6) > 0$	Saddle point
$6 > 0$	$5 > 0$	$5^2 - 4(6) > 0$	Unstable Node (source)
-12 $<$ 0	$0 = 0$	$0^2 - 4(-12) > 0$	Saddle point
$12 > 0$	-4 $<$ 0	-$4^2 - 4(12) < 0$	Stable Spiral (sink)

Table 1: Fixed point classification using the Jacobian J

4 Discussion and Conclusion

Upon completion of the report we have become familiar with multiple different graphical methods, initially we began with vector fields and using the quiver and streamline methods. These tools allowed us to see the many different solutions possible for differential equations without actually solving these equations. We also used these methods as another way to visualise simple harmonic oscillators and damped harmonic oscillators. We then went on to observe different types of population models, two in particular the Verhulst model (a logistic equation) and the Lotka-Volterra model. We were able to observe both models graphically using quiver and streamline plots and altered the initial conditions to observe the affect each one had on the system.

The Verhulst model's graph allowed us to better understand how the equation describes population growth and observe the assumptions and short coming of the model, particularly if we were to keep expanding our time frame with the model it would no longer accurately represent population, as well as the fact that the model does not account for external factors affecting population.

The Lotka-Volterra model combats the failure of the Verhulst method to account for external factors by primarily focusing on the affect two different species have on the population of each other. We used the example of foxes and rabbits to show that hunters have an inherent affect on the population of its pray and vice versa. This model wasn't perfect as we found that it would always repeat the pattern determined by the plot over time as the population never truly remains at zero. This is one of the big limitations of this model. All of this analysis would not have been possible without the use of quiver and streamline plots to visualise these relationships.

Going one step further with this repeat behaviour concept we explored limit cycles from non-linear differential equations. This showed us that some differential equations mapped to systems which either moved towards being in a stable state and entering a close phase diagram loop or an unstable state where the phase diagram would not be a closed loop. These different outcomes are controlled by whatever initial conditions are chosen for any particular set of non-linear differential equations.