

Classifying Equilibria

Recall that an equilibrium point of a differential equation is a point at which the derivative is equal to zero and therefore no change is taking place. Equilibria can be stable or unstable (or, in rare cases, semi-stable). In this lab, you will write a program that will determine the stability of equilibria.

Studying the Allee Effect

This lab will focus on the population model $N' = rN(1 - \frac{N}{K})(\frac{N}{A} - 1)$. This model describes what's known as an Allee effect, in which a population will only grow if it's above a certain size, A . Such effects can occur in populations of animals that breed or hunt cooperatively.

Exercise 1. Pick numerical values for r , A and K (K must be larger than A) and define a mathematical function for N' in SageMath.

Exercise 2. Find the equilibria of your differential equation and make a list of them, from smallest to largest. Assign the list to a variable.

An equilibrium point can be stable or unstable (or, occasionally, semi-stable). If an equilibrium point is stable, the system returns to it after a small perturbation. If it is unstable, even a tiny perturbation will send the system to a different equilibrium point or a trajectory of infinite growth.

Exercise 3. Simulate your differential equation for several different initial values. Use the results to determine the stability of the equilibria.

Which of these things happens depends on the vector field around the equilibrium point. If the change vectors point toward the equilibrium point (think of this as nudging the system state toward it), the equilibrium is stable. If they point away from it, the equilibrium is unstable. Finally, if the change vectors on one side of the equilibrium point toward it and those on the other side point away from it, the equilibrium is semi-stable. Fig. 1 illustrates these situations.

These observations can be rephrased in terms of the signs of N' . If N' is positive below an equilibrium point and negative above it, the equilibrium point is stable. If N' is negative below an equilibrium point and positive above it,

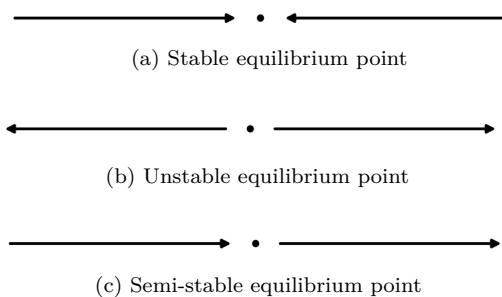


Figure 1: Different types of equilibria in one dimension. What else could the vector field around a semi-stable equilibrium point look like?

the equilibrium is unstable. Finally, if N' has the same sign on both sides of an equilibrium point, the point is semi-stable.

Exercise 4. Translate the above paragraph into an if-else statement in pseudocode. Include print commands.

We now face the problem of determining the sign of N' . One simple way to do this is to pick a number above or below the equilibrium of interest and close enough to it that there are no other equilibria between the equilibrium point and the test point. We then compute N' for that value of N , since N' can't change signs between equilibria. (Why?)

Exercise 5. Use your pseudocode from the previous exercise to write a Python function that classifies equilibria as stable, unstable or semi-stable. The function's input should consist of the equation for N' (given as a mathematical function) and two values of N , assumed to lie above and below the equilibrium point being studied. HINT: Remember that to use functions as inputs to other functions, just use their names – f rather than $f(x)$.

Multiple equilibrium points and linear stability

You now have code that can test the stability of a single equilibrium point. However, most differential equations have multiple equilibria.

There are a couple of different approaches we can follow to handle systems with multiple equilibria. One is to use the method of test points for each equilibrium point. This has the advantage of always working but is rather finicky in handling the largest and smallest equilibrium of a model. The other is to use Sage's symbolic capabilities to implement linear stability analysis. This has the disadvantage of failing when $\frac{df}{dN} = 0$, but this situation is rare. We will therefore pursue the second option.

Exercise 6. Using the `diff` function, find the derivative of N' and make it into a math function that takes N as an input. HINT: You can do this in one line of code.

Exercise 7. Using the function you just made, write a script that tests the stability of a single equilibrium point. If the derivative is 0, the script should print "Test failed".

Exercise 8. Modify your script so it will work on a list of equilibria, testing each in turn.

Exercise 9. Since nothing in your script refers to a particular model, it's fairly easy to turn it into a Python function that will perform stability testing for the equilibria of any differential equation. Copy your code and do this. The input to the function should be the name of a differential equation (this is just `f`, not `f(x)`) and a list of its equilibria, so the function should automatically compute the derivative of the given differential equation. Test your function on the logistic equation.