Searching for Bifurcations

Bifurcations of equilibria occur when a change in a parameter leads to a change in the number or stability of equilibria of a system. Among other things, bifurcations can enable cells to turn protein production on or off, lead to insect outbreaks and generally cause sudden changes of a system’s state in response to small changes in parameter values. In this lab, you will develop a program to search for parameter values at which bifurcations take place. In order to do this, you will need a new kind of loop called a while loop.

While loops

A for loop repeats an action a fixed number of times. However, sometimes you may not know how many times an action needs to be performed. Rather, you know that the action needs to be repeated until some condition is met. For example, the loop below prints numbers until it finds one whose square root is 2.5 or greater.

```python
n = 0
while sqrt(n) < 2.5:
    print n
    n = n+1
```

Notice that we needed to give the variable n a starting value. Also, note that the last line of the loop increases n by 1. If that line wasn’t there, we would have an infinite loop – a loop that never ends. This is a common mistake in programming. To interrupt an infinite loop in Sage, click the Stop button (the orange square), fix your code, and then run it again.

Sometimes, either a for loop or a while loop can perform a task. For example, this while loop prints the numbers from 1 to 10, which could easily be done with a for loop. (How?)

```python
n = 1
while n <= 10:
    print n
    n = n+1
```

We can use a while loop to search for numbers that meet certain criteria. The following example adds numbers until their total reaches 100 and then prints the last number.

```python
total = 0
n = 0
```
while total < 100:
    total = total + n
    n = n+1
print n

Exercise 1. Type in and run this example. Write a comment on every line that explains what it does.

Exercise 2. If you add up integers like in the example above, is the number you obtained in the previous problem necessary for the total to reach 100? (HINT: You may want to use a smaller target number, like 11, to find out.)

Exercise 3. Write a while loop that finds the smallest positive integer for which the product, rather than the sum, of all smaller positive integers is 100 or more.

Finding bifurcations

The differential equation $x' = \frac{x^2}{1+x^2} - rx$ can be used to model biological switches. As $r$ decreases from a high value, this system goes from having one equilibrium to having three (Figure 1).

Exercise 4. Simulate the model for three different values of $r$, using both a high and a low initial value of $x$ for each. Describe your results.

We want to know the value of $r$ at which the number of equilibria changes. Finding these values given a value of $r$ involves solving a cubic equation, which we’d rather avoid doing. Fortunately, SageMath has a very useful function, solve, for solving algebraic equations. For example, suppose we want to solve the equation $x^2 - 3x + 2 = 0$ for $x$. The syntax is:

```python
>>> solve(x^2 - 3*x + 2==0, x)
[x == 1, x == 2]
```

Notice that you have to use a double equal sign. Also, all symbols must be symbolic variables.

Figure 1 suggests that there is a value of $r$ for which $x' = \frac{x^2}{1+x^2} - rx$ has exactly two equilibria. Above this value, the system has one equilibrium and below it, it has three. This is the critical value of $r$ we are looking for.

One way to find this value is simply to make a guess and then see how many equilibria the system has for that value of $r$. If it has one, the value is too high and the next guess should be lower; if it has three, the next guess should be higher. This guess and check procedure allows us to approach the critical value of $r$ as closely as we like.

The phrase “as closely as we like” is crucial. It is very unlikely that the procedure outlined here will hit $r$ exactly, although the if-statement you write
should provide for this eventuality. Rather, as the highest and lowest possible values get closer and closer together, they become so similar that we can approximate the critical value of $r$ with their average. The idea is that we don’t care what’s going on in the twentieth decimal place and can, for most purposes, make do with an answer that’s “close enough”.

You will now outline and write a program that will find the value of $r$ at which the number of equilibria of $x’ = \frac{x^2}{1+x^2} - rx$ changes. (You can take advantage of the fact that the solve function outputs a list of values.) As you plan your code, consider the following questions.

- What might be good highest and lowest numbers to start with? (HINT: Use what you already know about the model.)
- Given highest and lowest possible values for $r$, what value should the program guess next?
- What variable should change if the guess is too high? If it’s too low?
- How can you make sure that guesses that are too high and those that are too low are treated the same way?
- How will you decide if your estimate of $r$ is good enough for the program to stop? Be careful to avoid coding an infinite loop, although you can just interrupt the calculation if you do.
Exercise 5. Write pseudocode for your program, including all important steps. The more thorough you are here, the easier the actual coding will be.

Exercise 6. Turn your pseudocode into a working program and use it to find an approximation to the critical value of $r$. Use plotting to check if your answer makes sense. Test your code on initial guesses that are both too high and too low and make sure it works in both cases. NOTE: SageMath solves equations in the most general way possible, which can give negative or complex solutions that are irrelevant to us. Therefore, your code must start with the commands `assume(x>=0)` and `assume(x, "real")`. HINT: Putting in `print` statements to view values of variables or see the sequence of actions a program performs is a very useful debugging method.

Exercise 7. Use your code to find the values of $r$ at which a bifurcation takes place for $x' = \frac{2x^2}{1+x^2} - rx$ and $x' = \frac{1.5x^2}{1+x^2} - rx$. Then, run simulations to confirm that bifurcations really do take place at these values.

Exercise 8. Modify your code so that it keeps a list of successive guesses and plot them. What do you observe?

Exercise 9. Modify the code from the previous exercise to make it demand a more precise estimate of $r$ and run it. Do this several times with the same initial highest and lowest values. Describe how the number of guesses the program makes changes with the requested precision. (You may want to make a table in a spreadsheet.)