# Bifurcations and Chaos in Mathematica

Chloe Wawrzyniak

Summer 2018

## 1 Modeling Bifurcations

Consider the ODE we studied this morning:

$$\frac{dy}{dt} = r - y^2$$

Use what you learned this morning and in yesterday's class to solve this ODE for the following $r$-values. For each one, you should solve with initial values corresponding to the constant solutions and initial values between each constant solution. For example, in the morning we learned that for $r = 9$, there are two initial conditions which give constant solutions: $y(0) = \pm 3$. So, you would solve the ODE corresponding to $r = 9$ five times, with five different initial conditions: 3, -3, a number less than -3, a number between -3 and 3, and a number greater than 3.

1. $r = -1$
2. $r = 0$
3. $r = 4$

For each $r$-value above, plot all of the solutions that you have for different initial conditions. Namely, you should have 3 graphs (one for each $r$-value), each with the same number of functions graphed on it as the number of initial conditions you started with. Don't forget to add a legend to your plot (note that you may need to use Evaluate for the plot legends command to work).

## 2 Systems of ODEs

Some applications involve multiple functions and their derivatives which are all related to eachother in some way. In that case, we may have not only multiple functions but also multiple equations to solve. We call just a setup a **System of Differential Equations**. At this point, we are still working with functions of one variable, so we can more specifically call these setups systems of ODEs. The syntax for solving an ODE in Mathematica is the same as the syntax that you used when including an initial condition in your DSolve function. Try solving the following system of ODEs using Mathematica and plotting the results.

$$\frac{dy}{dt} = y + 2x + t$$
$$\frac{dx}{dt} = 2y - 2x + \sin(t)$$

Don't forget that your solution involves two different functions, so you'll need to assign two functions to the solution values and then plot both of them on the same graph.

In the above example we worked with two functions and their derivatives. There's nothing to say we have to stop at two! Try solving the following system and plotting the solutions:

$$\begin{aligned}
\frac{dx}{dt} &= -\frac{1}{2}x \\
\frac{dy}{dt} &= \frac{1}{2}x - \frac{1}{4}y \\
\frac{dz}{dt} &= \frac{1}{4}y - \frac{1}{6}z
\end{aligned}$$

## 3 Chaos

One of the most interesting systems of ODEs that we're going to look at in this class was developed in 1963 by Edward Lorenz to model atmospheric convection (think: predicting the weather). The system now commonly known as the **Lorenz system** is as follows:

$$\begin{aligned}
\frac{dx}{dt} &= \sigma(y - x) \\
\frac{dy}{dt} &= rx - y - xz \\
\frac{dz}{dt} &= xy - bz
\end{aligned}$$

where $\sigma, r$, and $b$ are constants which depend on our intial conditions. We're going to focus our attention on the values $\sigma = -3$ and $b = 1$. In this case, when we're considering only $r$ as a parameter, the system has a bifurcation point around 26. We will (for the moment) impose the initial conditions $x(0) = z(0) = 0$ and $y(0) = 1$.

1. Use Mathematica to solve the Lorenz system with the above parameter choices and initial conditions and with $r = 10$ and plot the solutions. Note that you'll need to use NDSolve instead of DSolve because this system doesn't have an explicit solution (ie we can't write down the function explicitly).
2. Do the same thing but with $r = 20$. How does this change our solution?
3. Now do the same thing but with $r = 26.5$. How does this compare to your earlier solutions?

Have you ever heard of the butterfly affect? It's the idea that very small changes (like a butterfly flapping its wings in Brazil) can have major consequences (like a tornado in Texas). This idea has come up numerous times throughout history, and the term was coined by our very own Edward Lorenz to describe how small changes in the initial data propagate to have massive changes in the solutions to his system of ODEs.

4. Using $r = 26.5$ as we did in the third problem above, create a new function which corresponds to the initial conditions $x(0) = z(0) = 0.001$ and $y(0) = .999$.
5. Using $r = 26.5$ as we did in the third problem above, create a new function which corresponds to the initial conditions $x(0) = z(0) = 0.01$ and $y(0) = .99$.

6. Using $r = 26.5$ as we did in the third problem above, create a new function which corresponds to the initial conditions $x(0) = z(0) = 0.05$ and $y(0) = .95$.

7. Using $r = 26.5$ as we did in the third problem above, create a new function which corresponds to the initial conditions $x(0) = z(0) = 0.1$ and $y(0) = .9$.

8. Plot all four of the above functions, as well as the one you created in problem 3, on the same graph. What do you notice about the behavior as the initial conditions get further away from the starting values of $x(0) = z(0) = 0$ and $y(0) = 1$?

## 4 Challenge: Chaos and Secret Messages

Suppose you want to send a recording of a secret message, but you want only the recipient of the message to be able to read it. So, you would want to encode that message in such a way so that only the recipient has the key to decode it. One way to do that is to apply a filter which renders the message impossible to understand and which only your recipient can remove.

We just discussed how sensitive to initial conditions the solutions to the Lorenz system are. We can use that to our advantage to create a filter for sending secret messages. Remember, sounds can be represented on a computer by waves. So, our strategy for sending a secret message could be to represent our message as a wave, then adding a chaotic solution to the Lorenz system. The resulting message would be impossible to understand, and in order to remove the filter, one would have to know exactly what initial conditions you used to create the filter. Therefore, as long as your recipient has the right initial conditions, they can decode your message.

Mathematica can play and graph sounds and sound waves. So, your challenge is to encode a sound using the chaos we've modeled above, send your secret message to a friend who knows your initial conditions, and have them decode the message.