

March 31, 2015

This repository was created as part of my learning how to simulate dynamical systems. I believe once I'm confident in using `scipy` to simulate dynamical systems two things will happen

1. I will have gained intuition into how dynamical systems behave/a deeper grasp of the concepts involved and how equations can be manipulated to produce new dynamics
2. The power to simulate dynamical systems will enable me to experiment as I read the textbook on dynamical systems and chaos. I believe experimentation is the key to actually understanding something because it forces you to self-validate your hypotheses and is the same pipeline real scientists and mathematicians use to create new ideas. Additionally, in the first stages of learning complex concepts actively (through play and experimentations), you are forced to distill the complex concepts into their purest essence – and hereby you make them yours.

I'd also like to point out that what I am capable of doing can be described by the equation $\dot{x} = x^2$ (I could also describe the whole system of learning as a dynamical system). At the beginning, things are new and you don't really have experience dealing with such complexity. As time passes and you do more reading, and the ideas start to marinate in your mind, new structure and understanding is created. Then, you begin to see connections and start to form hypotheses, questions, and more nuanced observations. The more things you learn, the more connections exist and the more experienced you are. Finally, you hit the singularity point and reach polynomial growth. I'm currently hitting this point, where everything I've learned is coming together and I finally start creating something of my own, and that is all because of the tutorial I'm following.

I become a power dynamical systems modeller and I become more powerful in general. So, let's go model dynamical systems!

Realizations

As I go through the tutorial, I realize just how powerful dynamical systems of differential equations are. I would also like to point out that this approach to learning math is very interesting - I learn just enough to be dangerous and to experiment, but not the formal stuff. I hope this won't come back to haunt me later on, though I'm trying to supplement informal exploration with more rigorous mathematical pursuits, such as doing the problem sets in Knuth's concrete mathematics. Another thing to point out is the value in computational experimentation with the mathematical techniques I learn. Dynamical systems are easy to experiment with because there is an intuitive relationship between the system you create and the phase plots the solver generates. And dynamical systems also happen to be really fun. Nevertheless, the fun stops here; things need to get formal, and I need to start solving really difficult problems, exponentially more difficult problems. To do that optimally requires a deep understanding and great skill and intuition of the wide array of tools that can be used to solve a given problem. You can't play around with linear algebra, set theory, probability theory the same way you can play around with dynamical systems. \

The approach I've developed when I encounter new concepts/equations in literature is to step back and consider (1) what brought them to be, and (2) their implications under different inputs.

For example: say you've got two neural spike trains and want to cross-correlate them. Let's use the cross-correlation algorithm discussed in the [Feldt. literature] [1] i.e. just the dot product of the two spike trains.

Let's look at what the dot product does (using this pseudocode):

```
sum(a,b) = sum i over a[i]*b[i]
```

I know this is a very simple example, but I'm trying to illustrate my technique. We can start asking questions:

- When $a[i] < 0, b[i] > 0$ or [?]
- When $\text{sign}(a[i]) == \text{sign}(b[i])$?

[1]: Deciphering the Neural and Molecular Mechanisms of *C. elegans* Behavior