Problem Set 2

In this problem set we will take a deeper dive into one-dimensional flows, bifurcations, and tipping points. Recommended reading: Chapters 2-3 of Strogatz.

Let's return to the ecological example from class. One other way the per-capita growth rate of a population N may change with N is parabolically:

$$\frac{dN}{dt} = rN(-N^2 + aN - b), \tag{1}$$

where a, b, and r are positive constants. For high enough N, we get a declining per-capita growth rate, similar to the logistic model we discussed. However, for low N we get an *increasing* per capita growth rate: this is called an "Allee effect" and typically reflects the benefits of cooperation.

1. (a) Plot or sketch dN/dt versus N, assuming (for this part of the question only) that $a^2 > 4b$. How many fixed points are there, and what is their stability?

   (b) Derive the fixed points (i.e. $N^*(r, a, b)$) analytically, and confirm the graphical intuition for stability/instability by Taylor expanding around them, as we did in class.

   (c) What kind of bifurcation can the system undergo as b is changed, and when does it occur? What do you think b might represent (ecologically speaking)[1]?

   (d) `allee` numerically integrates Eq. (1) for different initial values of N. Start with N at a few different values (above and below each of the nonzero fixed points works well), plot $N(t)$ in each case, and (briefly) discuss. As usual, please reach out if you have any issues running the code.

   (e) What would happen if we start at the largest-N fixed point and changed b until the system passes through the bifurcation? What would this mean for the population N?

2. In the real world we don't necessarily know how system parameters are changing, and thus whether or not a bifurcation is about to occur. However, there are other ways to know a bifurcation is near. Here, we'll develop one such early warning signal: *critical slowing down*.

   (a) Consider the saddle-node normal form:

$$\frac{dx}{dt} = r - x^2. \tag{2}$$

[1] You don't need to have studied ecology. Anything reasonable that is compatible with the mathematical behavior will get full credit.

This has a stable fixed point at $x^* = \sqrt{r}$. As we saw in class, small perturbations from a stable fixed point decay exponentially: $\eta = \eta_0 e^{-t/\tau}$. How does the characteristic recovery timescale (τ) depend on r? What happens as the bifurcation is approached ($r \to 0$)?

(b) Now consider the model in Eq. (1) again. Show that the characteristic timescale for the largest-N stable fixed point is given by

$$\tau = \frac{1}{-r\left(-\frac{a^2}{2} + 2b - a\sqrt{\frac{a^2}{4} - b}\right)}. \tag{3}$$

(c) Rewrite τ in terms of a new parameter β that depends linearly on b and is zero at the bifurcation. In other words, β becomes our new bifurcation parameter. What happens in the small-β limit? Comment on any similarities or differences with the answer to part (a).

(d) A more realistic version of Eq. (1) might contain some constant-amplitude random fluctuations. Given what we now know, what do you think might happen to these random fluctuations as the bifurcation is approached that could serve as an early warning signal?

(e) `allee_random` is precisely an implementation of Eq. (1) with such random fluctuations[2]. Here, b is set up to linearly change with time, and you have control over how far and how fast it changes. Use the code to simulate the approach to the bifurcation, and find the early warning signal.

The effect is a little subtle to see by eye (usually it's quantified through formal quantitative analysis of the time series), but it should be clearly visible if you know what you're looking for. Since there is randomness involved, you might need to run the code multiple times for the same parameter values before you see something convincing.

[2] The implementation is a little crude (rigorous integration of ODEs with random terms is a complex topic beyond the scope of this course) but sufficient for our purposes. We recommend not changing the "timestep" parameter, but if you do, be warned that you may get spurious results if it's not small enough. Also, the Matlab version of the code requires you to have the Statistics and Machine Learning toolbox installed; as usual, reach out if you have any issues.