Forced Van der Pol Oscillator

Deliverable I:

For the first deliverable, I used three main parts of code: an ode solver (ode45), two single order differential equations as a function handle, and the main script. One key feature which came into play in deliverable II was the use of ε as a global variable. By establishing as a global variable, my function would be able to use that to solve the odes. From here, I used the following decomposition of the second order differential equation into these two first order equations:

$$x' = y - \mu * \left(\frac{x^3}{3} - x\right) \qquad \text{Equation 1.1}$$

$$y' = -x + b * \cos(\omega t) \qquad \text{Equation 1.2}$$

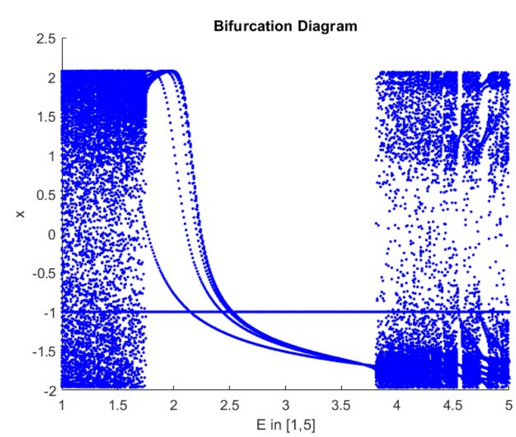
$$\mu = \frac{\varepsilon}{\omega} \qquad \text{Equation 1.3}$$

By passing these equations through our ODE solver, we obtain the solves values x, y, and t. From there, we can pass these values through Equation 1.1 in our main script to obtain $\frac{dx}{dt}$. Lastly, we can plot our x values from our ODE solver against our $\frac{dx}{dt}$ values to obtain a similar plot for the one provided in the problem statement.

Deliverable II:

Deliverable II was very tricky for me; I was unsure of how to produce the desired result but here is my thought process and attempt at the solution. For a bifurcation diagram, I wanted to extract the repeated periodic motion of the phase plane solution.

1. Begin looping over ε values from 1 to 5
2. Our loop will run our ODE solver for each ε value to obtain x, y, and t values
3. Since we set the period to 10 seconds ($\omega = \frac{2\pi}{10}$) we should see values become repeated every 10 seconds if the value of ε is stable, so lets have our time array jump in 10 second intervals
4. Lets plot our x values for each of the ε values over the given period



I played around a bit with the initial values of x and concluded that [-1 1] was the most similar (or at least visually in the location of the gaps) to that presented in the problem statement. The instrumental piece here was the use of the global variable. By using ε as a global variable, we were able to vary ε in our VanDerPol function which allowed us to use a for loop for the entire process. Another thing to note was the use of a time array which jumped in 10 second intervals. This was done to jump in periodic places where we should expect the location to remain equal. One issue I ran into was uploading all the files to gradescope so here are all of my files: https://drive.google.com/drive/folders/1BEMmYnWsXbbb5f-AYRhV1gy0uHgeaehs?usp=sharing