The task, overall

Your task is to formulate the problem of *discovering the governing equations of a dynamical system from data* as a program synthesis problem.

A few considerations

- While this is a practical task in which we ask you to program something, we want to see (1) how you think and (2) that you can turn things into code. We don't care about the code quality, the performance, and such. Don't spend time on that.
- You don't have to do everything from scratch. If there are libraries that achieve parts of the functionality you need, feel free to use them. If you feel more comfortable doing everything from scratch, that is also fine. A few things we can immediately point out:
  - A dynamical system will have some numeric parameters that need to be fit to data. Feel free to use any library to do it for you
  - you can make use of the following program synthesis libraries: Herb and DeepSynth. We only ask you not to use any synthesiser out of the box, but show us that you understand how they work.
- The assignment is intentionally underspecified and you have the freedom to fill in the details however you like. Ultimately, we do not expect a specific solution from you. If you need more information about any part of the task, please don't hesitate to send us an email.
- There is no need to implement state-of-the-art techniques. Stick to simple ones; they will suffice for this problem.
- We will give you two systems to discover, but do not worry if you don't manage to discover them (it might as well be the case that this is too difficult). We want to see how your process.

Part 1: Program synthesis lecture

Watch this lecture by Nadia Polikarpova to understand the basics of program synthesis.
For more information, you can refer to the following course for resources.

Part 2: Generate some data

We want you to focus on discovering the following two systems.

System 1

*\dot{x} = x - 0.1 * x * y*

*\dot{y} = -1.5y + 0.075x*y*

System 2

\dot{x} = *1 − x − x**y/4 ,

\dot{y} = *(*2 * z − 1) * y,

\dot{z} = x/4 − 2 * y^3

\dot{x} represents a change in variable x in one time step. You can assume that time steps are discrete.

Your first task is to generate data that you can use to discover the system; this data will essentially describe the evolution of the dynamical system over time. You can choose how many steps of execution you want to generate and from how many different starting points.

Part 3: Make a simple program synthesiser

Part 1 should have given you an understanding of what program synthesis is. Part 2 gives you the data of the evolution of two dynamical systems.

In Part 3, we want you to implement a (simple!) program synthesis (following the first part of the lecture) and try to identify the dynamical systems from the data you generated. Your implementation should take in the data and any other input you deem necessary, and output the set of equations. Your approach should be able to explore both linear and polynomial dynamical systems of degree up to 3. There is no need to make it possible to discover more complex systems. You can assume that you always know the number of variables of a system: for System 1, you assume that you are looking for a system of 2 variables; for System 2, you know that there are 3 variables involved.

Document the decisions you made along the way. Which components of a program synthesiser do you need to specify? How did you make the choice for each of those components? How did you implement them? If you conducted an experiment and reconsidered your previous choice, document that and explain your reasoning to us. How many data points (different evolutions of the system, starting from different initial points) did you consider? There is no need to make an elaborate report; bullet points will suffice.

Do not stress if your implementation cannot discover the ground truth equations. It is not important that you manage to do exactly that; this will still show us how you think about the problems.

Task 4: Think further

For this part, you don't have to implement anything. These are just a few questions we want you to think about for the interview:

- What would you have to change in your method if your data would not contain the observation of every step of the system, but can have missing observations? For example, you see the values for *x* and *y* of System 1 in time steps (1,2,3,5,7) but not in 4 and 6.
- Assume that the data is now noisy, meaning that we do not observe the actual value of the variables but there is some small perturbation added to it (for example, by sampling a random value from a very tight Gaussian distribution centered at 0 and adding it to each variable of the system at every time step). What would you change about your method to minimise the impact of noise?

What to send us?

Simply send us your code (no need to overdocument it) and a brief report explaining how you approached the problem. Don't try to make it pretty, we will talk about it in the next interview.