# 1 Difference equations

Simulating models based on a difference equation is straightforward: simply apply the given equation over and over again, in an iterative way. After every step, the result needs to be stored in an appropriate data structure (list, array, ...).

As a light start into this tutorial this is ideal, as you can focus on familiarising yourselves with the programming environment, before getting into more complicated issues.

## A First-order difference equation: Cobweb model

In the first exercise, we are going to look at a simple first-order difference equation, the basic version of the Cobweb model from the first tutorial. The equation we recover from the model is

$$p_t = -\frac{\delta}{\beta} p_{t-1} + \frac{\alpha + \gamma}{\beta} \tag{1}$$

It is generally good practice in programming to separate different parts of the procedure, so it is easier to keep a good overview of the code. Code that is used multiple times should be wrapped in a function. In this case, a single line (the equation) suffices for the single time step and hence it wouldn't be so necessary to write a separate function for it. However, it still seems appropriate to break up the problem into smaller parts, at least for educational purposes.

### A.1 Exercise

Complete the function `Cobweb_1step` for one time step (i. e. from time $t$ to $t + 1$). It takes as arguments the current price, as well as the model parameters $\alpha$, $\beta$, $\gamma$, $\delta$, and returns the next period's price.

### A.2 Exercise

Now we have to iterate this equation several times in order to analyse the dynamic behaviour of the model. Write the for-loop that applies your 1-step function $T$ times ($T$ and the model parameters are defined below, feel free to play around with the values) and store the results in the results list (append the new price after every iteration).

You can verify your results by printing out the results, or by plotting the time series. The necessary codes are already provided in the script. Feel free to play around with values of model parameter to see how they change the dynamic behaviour of the model. Can you change them so the model becomes stable?

## B Second-order difference equation

Here we are looking at the equation of exercise 0 c) (ii) in the first tutorial:

$$y_t = 1.1y_{t-1} - 0.6y_{t-2} + 1100 \tag{2}$$

### B.1 Exercise

Complete the function `second_order_1step` that implements one time step of this model. Instead of implementing it strictly with the parameters 1.1, −0.6 and 1100, leave these as abstract parameters a, b, and c. That way, you can change your model parameters more easily and check how the model changes.

### B.2 Exercise

Now we want to apply this function iteratively to simulate the model's behaviour over time. I. e. write a for-loop that runs this model $T$ times.

Again, you can verify the results by plotting the resulting time series. The codes are provided in the script.

## C Bonus exercise

Implement and plot any other model form the first tutorial or the additional exercise, section 1. Plot the results to find out how it behaves.