

practical_3_positive_feedback_solutions

October 27, 2019

Name: Write your name here

1 SYSTEMS AND NETWORK BIOLOGY - PRACTICAL 3

2 Positive feedback

To submit your report, answer the questions below and save the *notebook* clicking on File > Download as > iPython Notebook in the menu at the top of the page. **Rename the notebook file** to "practicalN_name1_name2.ipynb", where N is the number of the practical, and name1 and name2 are the first surnames of the two team members (only one name if the report is sent individually). Finally, **submit the resulting file through the Aula ESCI**.

Remember to label the axes in all the plots.

*IMPORTANT REMINDER: Before the final submission, remember to **reset the kernel** and re-run the whole notebook again to check that it works.*

The objective of this practical is to explore the behaviour of a system with a positive feedback motif. In particular, we will use numerical integration of an ordinary differential equation model to analyze the behavior of a phosphorylation/dephosphorylation cycle as the kinase concentration (acting as input signal) is cycled from a low to a high level and back.

Consider the phosphorylation/dephosphorylation cycle shown in the figure below, in which phosphorylation is induced by both a stimulus signal S and by the phosphorylated protein itself, in the form of a positive feedback loop. Assuming that the latter process is cooperative, the model of this system can be written as:

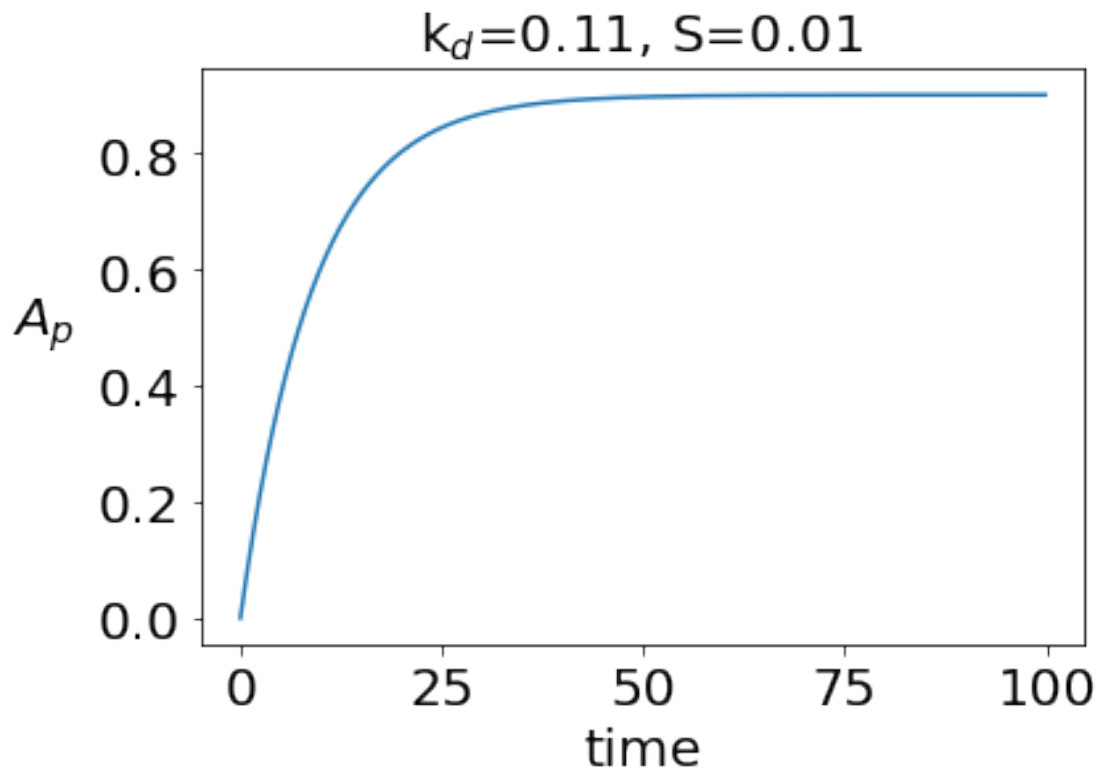
$$\frac{dA_p}{dt} = k_p S A + \beta \frac{A_p^n}{K^n + A_p^n} - k_d A_p$$

where k_p is the activation (phosphorylation) rate, k_d is the inactivation (dephosphorylation) rate, A is the concentration of the unphosphorylated protein, and A_p is the concentration of the phosphorylated protein. We will assume that the total protein concentration, $A_T = A + A_p$, is constant. We will consider the following parameter values: $K = 50$, $A_T = 100$, $k_d = 0.11$, $\beta = 10$, $k_p = 0.1$, and $n = 4$.

In [1]:

First, integrate numerically, up to $t = 100$, the differential equation above for a low (basal) stimulus level, such as $S = 0.01$, assuming that initially the protein is completely unphosphorylated.

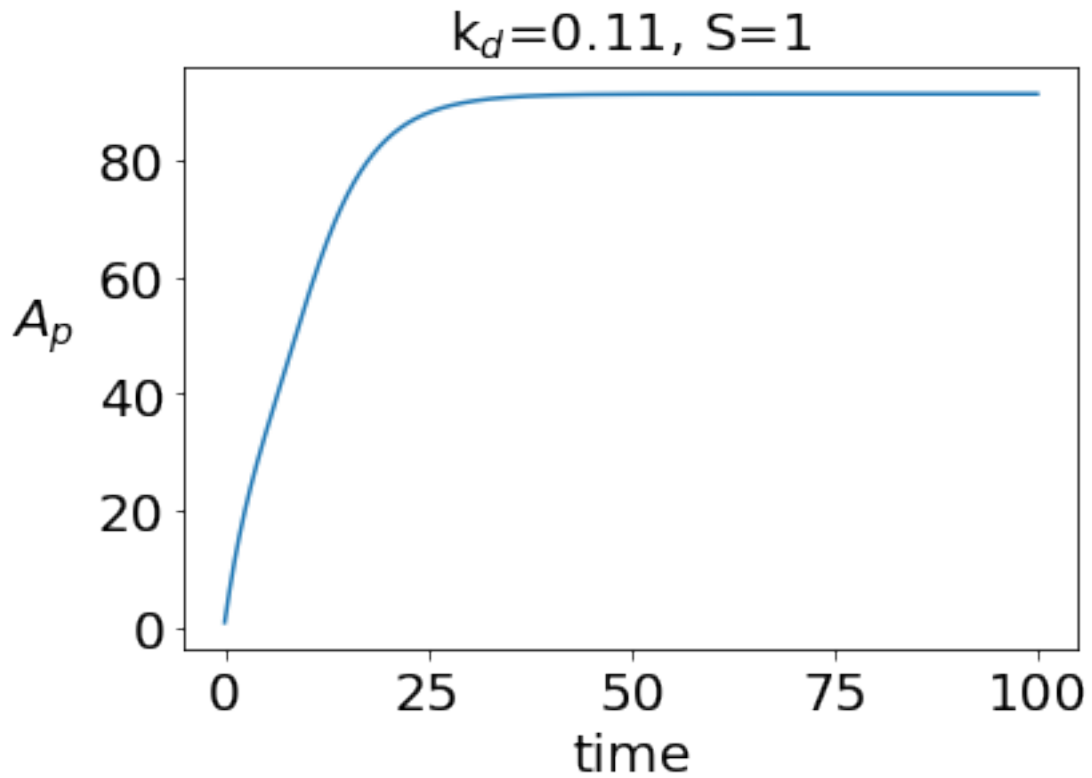
In [2] :



Comment your result here

Nest, starting from the stationary situation obtained in the previous case, study the effect of suddenly applying a large stimulus signal, so that S suddenly jumps to a high level, such as $S = 1$.

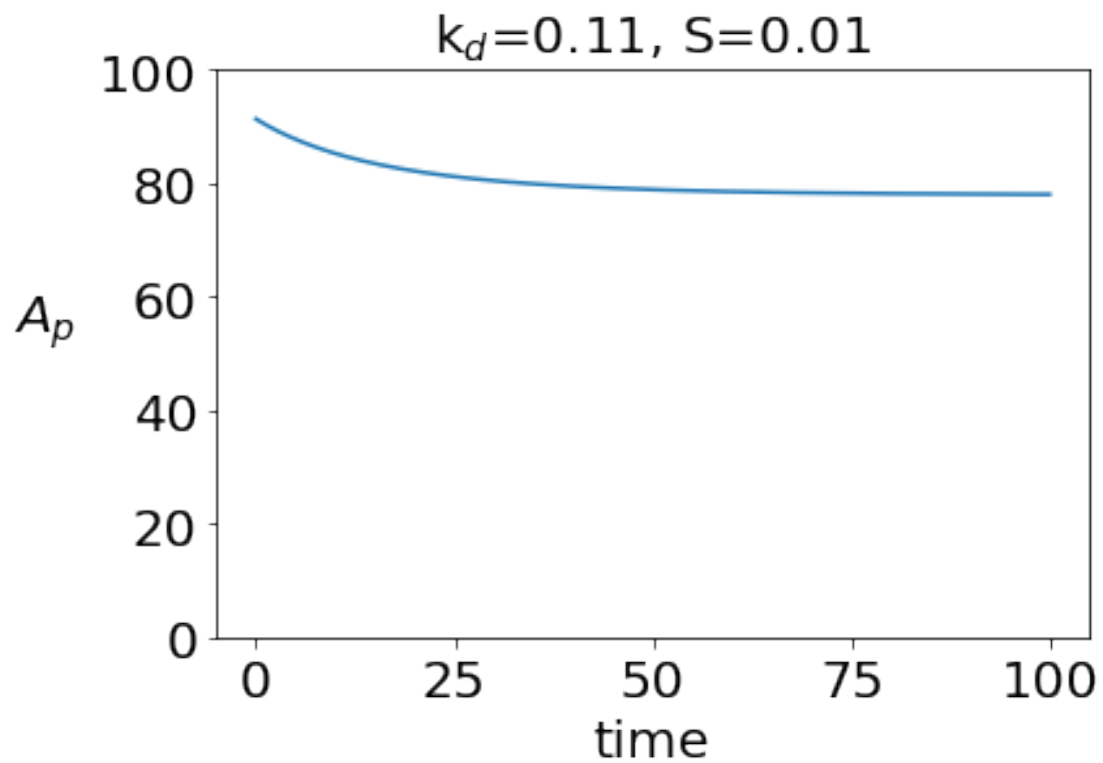
In [3] :



Comment your result here

Next, starting from the latest stationary situation, integrate the equations again for $S = 0.01$. Does the system go back to its original state?

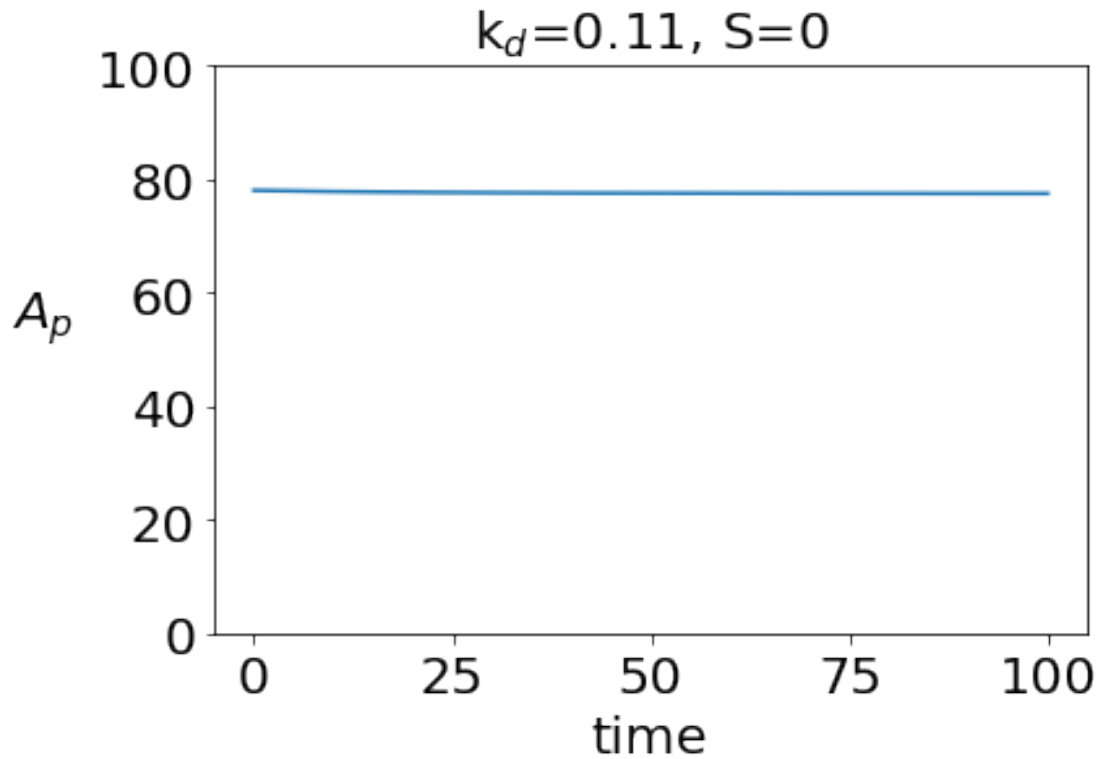
In [4] :



Comment your result here

Finally, continue further to $S = 0$, thus eliminating the stimulus completely. Does the system now go back to its original state?

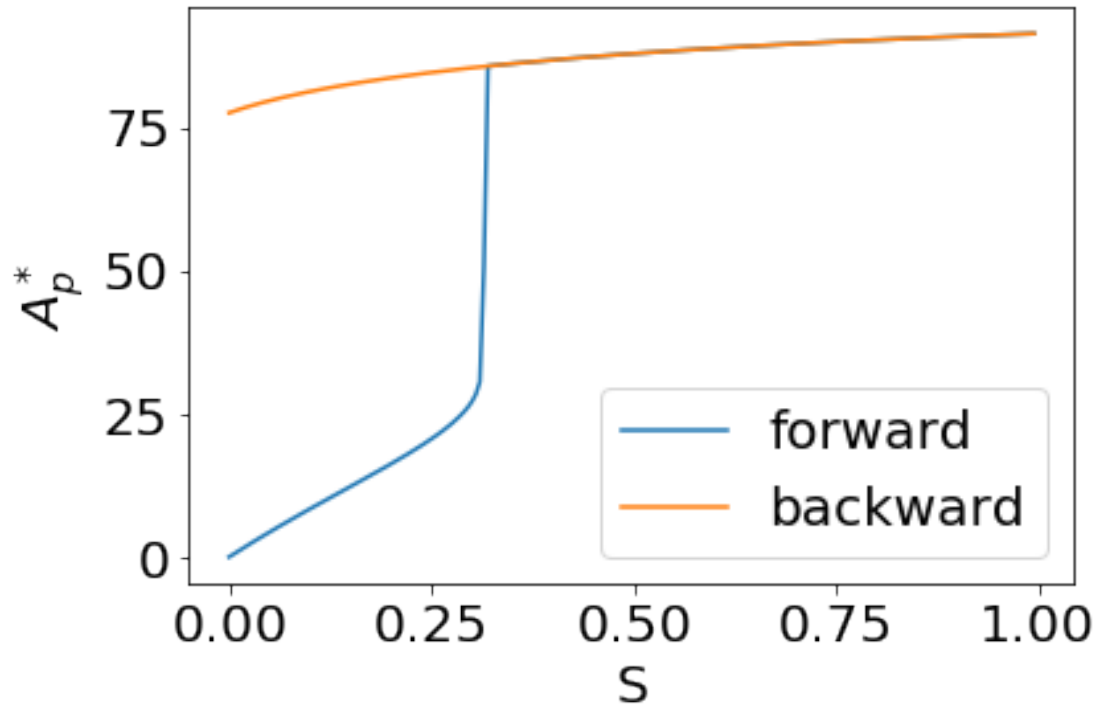
In [5] :



Comment your result here

Now we will manually compute a bifurcation diagram of the system. In order to do this, choose a range of S values from 0 to 1. Start for $S=0$ considering that the protein is unphosphorylated at the beginning. Then, for each S value do a simulation starting each integration at the previous steady state, and save the new steady state value. Plot the results in a graph where the x axis is S , and the y axis is the steady state reached (from each simulation). Then do the same but in reverse order (starting from high S) and plot the results in a different color.

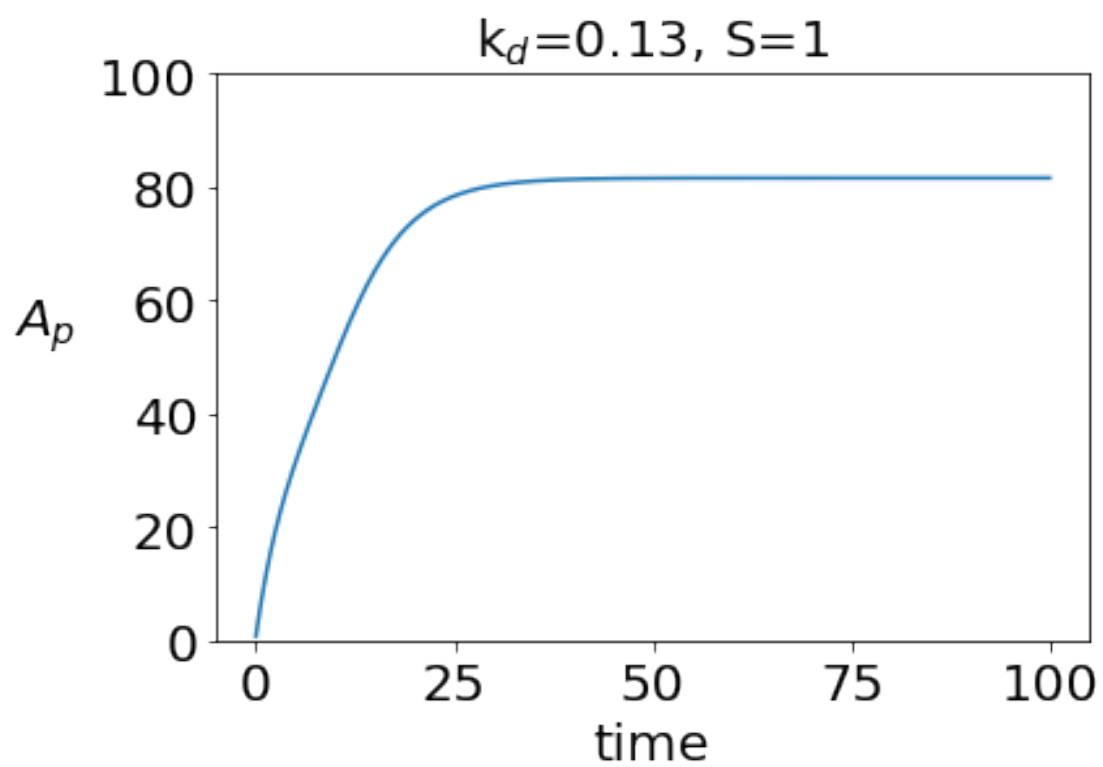
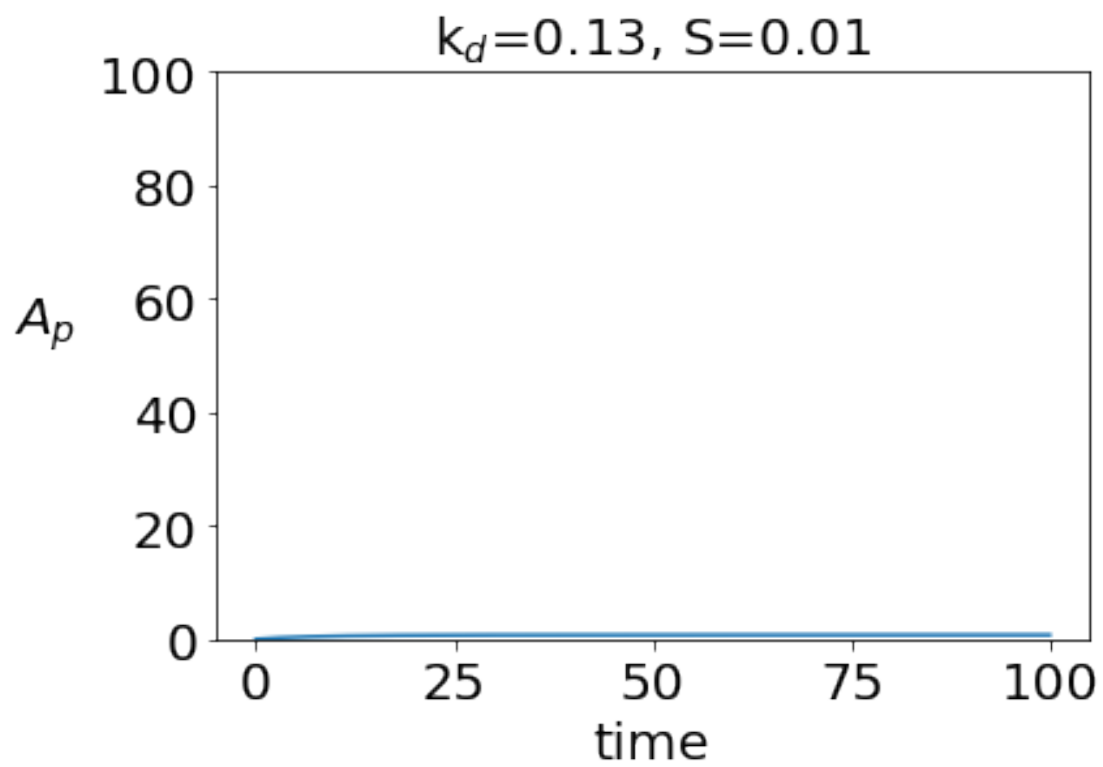
In [7] :

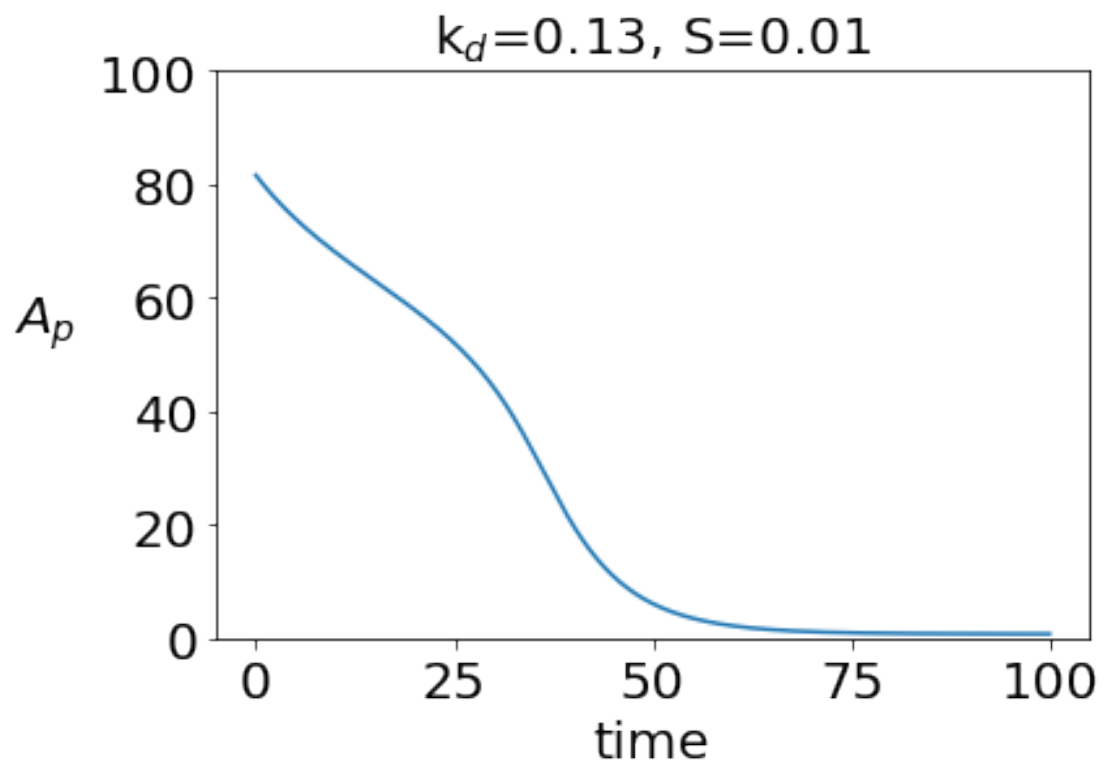


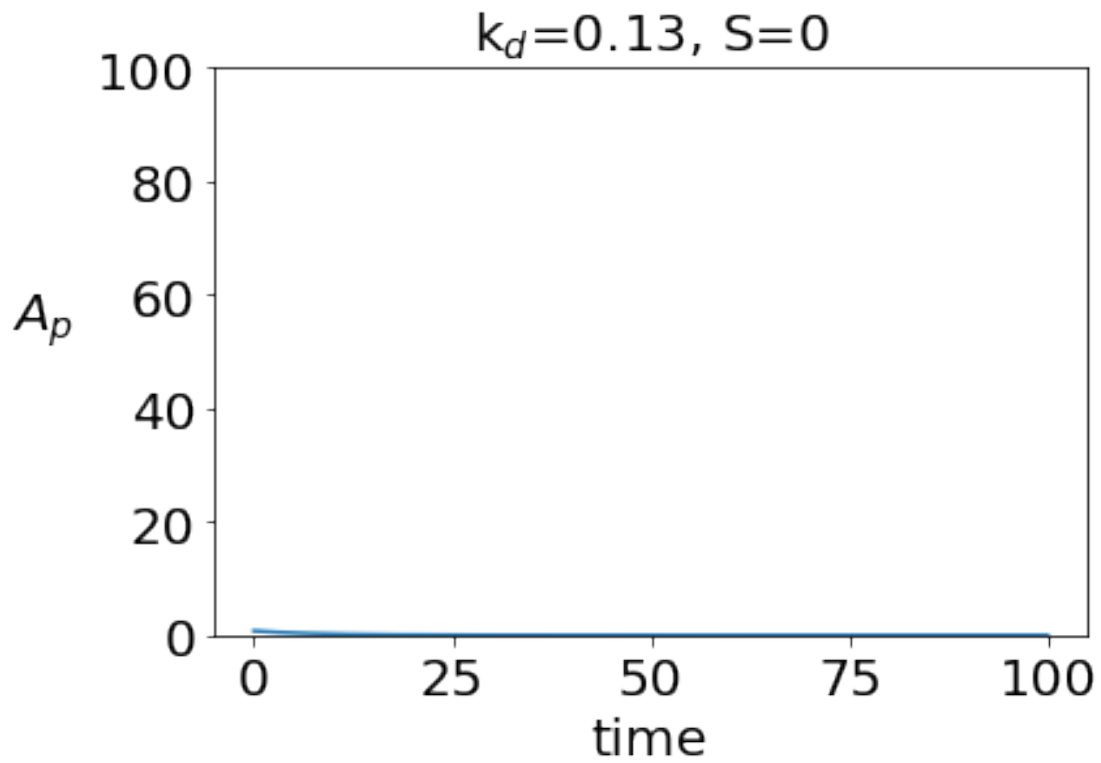
Comment your result here

Finally, repeat the analysis above for $k_d = 0.13$. First, integrate for fixed values of S (increasing first from $S = 0.01$ to $S = 1$, and then decreasing back to $S = 0.01$ and $S = 0$), starting in each case from the previous steady state. Comment on the differences with respect to the case $k_d = 0.11$.

In [8] :



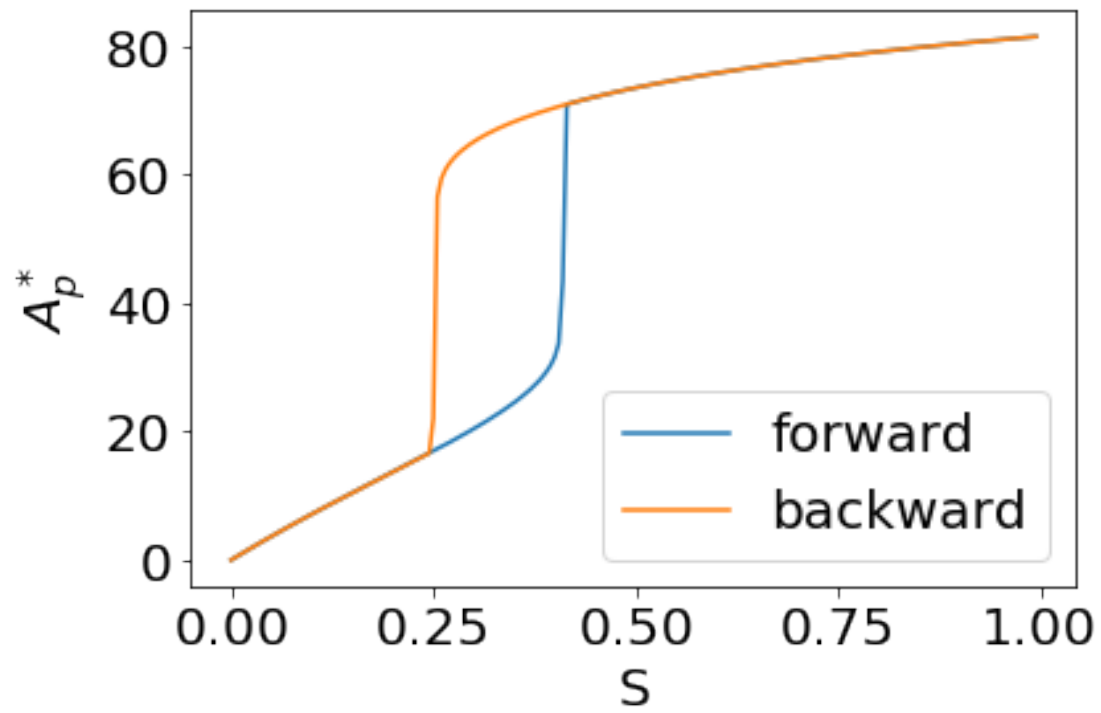




Comment your result here

Finally, calculate numerically the bifurcation diagram for $k_d = 0.13$ and compare it again with the case $k_d = 0.11$.

In [11]:



In []:

Comment your result here