Introductory Exercise: Stock Dynamics by hand

This little exercise will help you understand the principle of stock-driven models. It is **essential** that you understand this principle before you start programming this type of model in Python.

Stock-driven models

A stock-driven model is a model in which the stock (measured at the end of the accounting year) and the lifetime of each cohort are given; both as independent variables. Subsequently, the stock change, the inflow, and the outflow are calculated (dependent variables).

Task: Calculate the missing data in the table below. Assume that the lifetime of the products in the stock is 4 years. While filling out the table, reflect on the following:

- What **operations** do you conduct to calculate the different dependent variables?
- In what **sequence** do you conduct the operations to fill out the entire table (e.g., column by column, or row by row, or some kind of iteration…)?

	Annual stock (end of the year)	**Stock change** (during the year)	**Inflow** (during the year)	**Outflow** (during the year)
Year t	**S_t [tons]**	**dS_t/dt [tons/yr]**	**I_t [tons/yr]**	**O_t [tons/yr]**
t_0=2000	0	0	0	0
2001	100	100	100	0
2002	200			
2003	400			
2004	800			
2005	1600			
2006	3000			
2007	4000			
2008	5000			
2009	5500			
2010	6000			
2011	6000			
2012	6000			

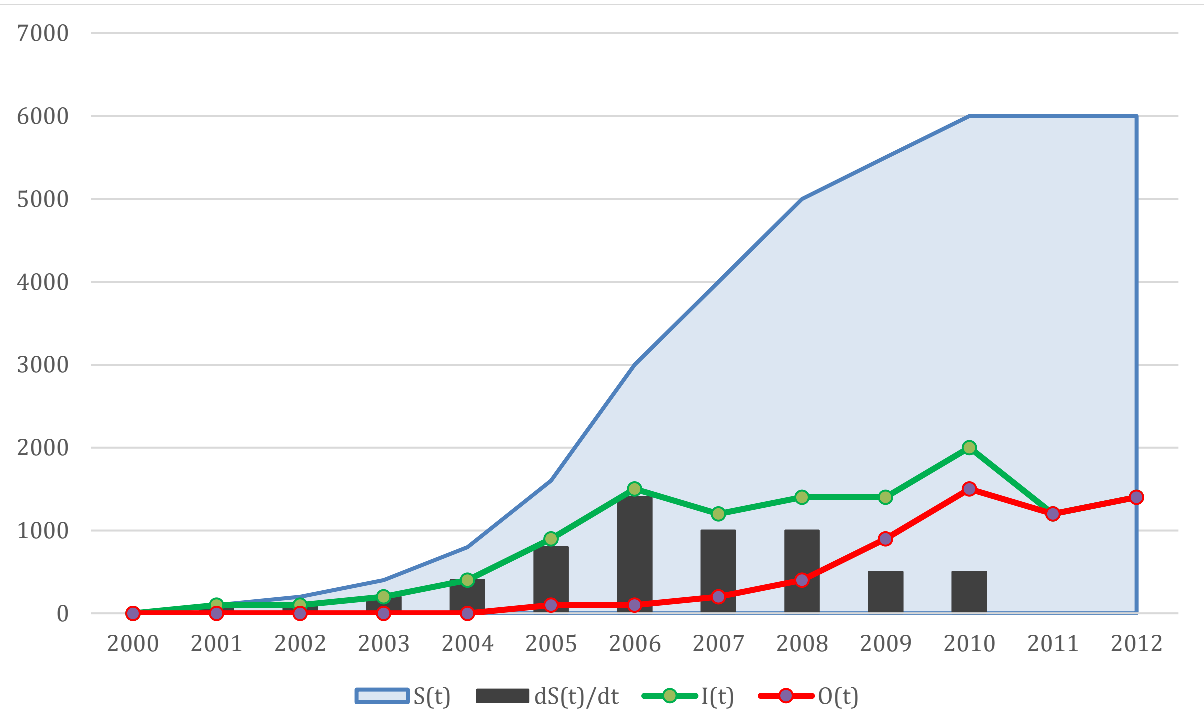
Solution and additional examples

1. Stock-driven model – 4 years sharp lifetime

This table contains the solution to the first example:

Year t	Annual stock (end of the year) S_t [tons]	Stock change (during the year) dS_t/dt [tons/yr]	Inflow (during the year) I_t [tons/yr]	Outflow (during the year) O_t [tons/yr]
t_0=2000	0	0	0	0
2001	100	100	100	0
2002	200	100	100	0
2003	400	200	200	0
2004	800	400	400	0
2005	1600	800	800+100 = 900	100
2006	3000	1400	1400+100 = 1500	100
2007	4000	1000	1000+200 = 1200	200
2008	5000	1000	1000+400 = 1400	400
2009	5500	500	500+900 = 1400	900
2010	6000	500	500+1500=2000	1500
2011	6000	0	1200	1200
2012	6000	0	1400	1400

We can represent the results on a graph:



The algorithm used to find the solution can be described as follows:
We repeat these steps year by-year, for each year, starting in 2001:

a. Calculate the outflow from inflows from all previous years and the lifetime distribution. In this simple example, O(t) = I(t-4), but the general rule would be :

$$\mathrm{O(t)} = \sum_{c=0}^{t} I(c) * \lambda(c,t)$$

With λ the lifetime distribution for each cohort.

b. Calculate the stock change between the current year and the previous year:

$$\frac{\mathrm{dS(t)}}{\mathrm{dt}} = S(t) - S(t-1)$$

c. Calculate the inflow:

$$I(t) = \frac{\mathrm{dS(t)}}{\mathrm{dt}} + O(t)$$

d. Go back to step a. for the next year, until we reach 2012.

We notice that in a stock-driven model, the calculation must be done row by row (year by year). We cannot do it column by column, because in order to calculate the outflows for each cohort, we need to know all previous inflows.

2. Stock-driven model – 3.5 years lifetime

In this new example, we now consider that for each cohort, half of the stock is retired after 3 years and the rest after 4 years. The average lifetime is then 3.5 years, and we can write the following equation for the outflows:

$$O(t) = \ 0.5 * I(t-3) + 0.5 * \ I(t-4)$$

How does this new hypothesis on the lifetime affect the results?

	Annual stock (end of the year)	**Stock change** (during the year)	**Inflow** (during the year)	**Outflow** (during the year)
Year t	**S_t [tons]**	**dS_t/dt [tons/yr]**	**I_t [tons/yr]**	**O_t [tons/yr]**
t_0=2000	0	0	0	0
2001	100			
2002	200			
2003	400			
2004	800			
2005	1600			
2006	3000			
2007	4000			
2008	5000			
2009	5500			
2010	6000			
2011	6000			
2012	6000			

3. Input-driven model – 4 years sharp lifetime

As in the first example, we assume a lifetime of exactly 4 years.
In this input-driven model, only the past inflows and the lifetime of each cohort are given.

How to modify the algorithm to solve the input-driven model? Is it still necessary to do a year-by-year calculation?

	Annual stock (end of the year)	**Stock change** (during the year)	**Inflow** (during the year)	**Outflow** (during the year)
Year t	**S_t [tons]**	**dS_t/dt [tons/yr]**	**I_t [tons/yr]**	**O_t [tons/yr]**
t_0=2000	0	0	0	0
2001			100	
2002			200	
2003			400	
2004			800	
2005			1600	
2006			3000	
2007			4000	
2008			5000	
2009			5500	
2010			6000	
2011			6000	
2012			6000	

Next step: how to write these algorithms in Python, using a for loop?