

Multiscale model of the different modes of invasion

This notebook is a companion for the article on a model describing the different modes of invasion using the tool PhysiBoSS.

PhysiBoSS is a modeling tool combining an agent-based approach for intercellular interactions and a Boolean model for the intracellular description.

The different parameters that are defined for the model show a variety of behaviors. By monitoring the values of some of the model variables, we show how the modes of invasion occur and evolve.

First, we define the libraries that are needed to perform the variable analyses

Entrée [1]:

```
import matplotlib
import matplotlib.pyplot as plt
import math
import numpy as np
from matplotlib.widgets import Slider, Button
from ipywidgets import *
%matplotlib notebook
```

Analysis of the evolution of the variables of the model

All the functions that allow cell interactions with the ECM, adhesion and evolution of the variables, have been written for this model ad hoc.

We focus on our analyses on a subset of parameters that regulate the main cell behaviors: adhesion, motility and degradation of the ECM. Their values range between $[0, 1]$, since they all represent a percentage of activation:

- pintegrin: this parameter represents the percentage of integrins on the cell surface. The higher this value is, the more ECM the cell is able to degrade.
- pmotility: this parameter represents the tendency for a cell to move towards a certain direction. The higher this value is, the bigger the steps of the cell during migration will be.
- padhesion: this parameter represents the percentage of cell junction located on the surface of the cell membrane. This value can also be seen as a percentage of the amount of E-Cadherin present. In the modeling framework, depending on this value and the threshold selected by the user, the cells will develop more or less hook-like junctions and this effect will be linked to the possibility to form clusters of cells.

These variables evolve in time and are function of the state of the surrounding. They are based on the following equation:

$$X_{t+1} = X_t + X_t(1 - X_t) \frac{d_t}{650}$$

Where (X_t) represents the value of the coefficient at step t, X_{t+1} at step t+1, and d_t the phenotype step.

An example for the phenotype ECM_adhesion and the variable pintegrin

The phenotypes of individual cells will be modified according to these variables. As example, let us consider the read-out "ECM_adhesion".

At each step of the simulation, the value of this output node is checked and the equation for the evolution of the associated variable is called (in this case the variable is "pintegrin"). The value of the phentoype is used to set the sign of d_t . If the value of the node is 1 (activation), d_t will have positive sign, if the value of the node is 0, d_t will have negative sign. According to the sign, the curve associated to the equation changes its concavity, increasing or decreasing the value of the variable.

Below we report the evolution of the curve for both situations.

Equation with negative d_t .

Entrée [7]:

```
# Initial values are set for a negative dt

init_x = 1
dt = -6
time_step = np.arange(0, 650)

def update_param(x, time_step, dt):

    values = []

    for i in time_step:

        x = x + (x*(1-x)*(dt/650))

        values.append(x)

    return values
```

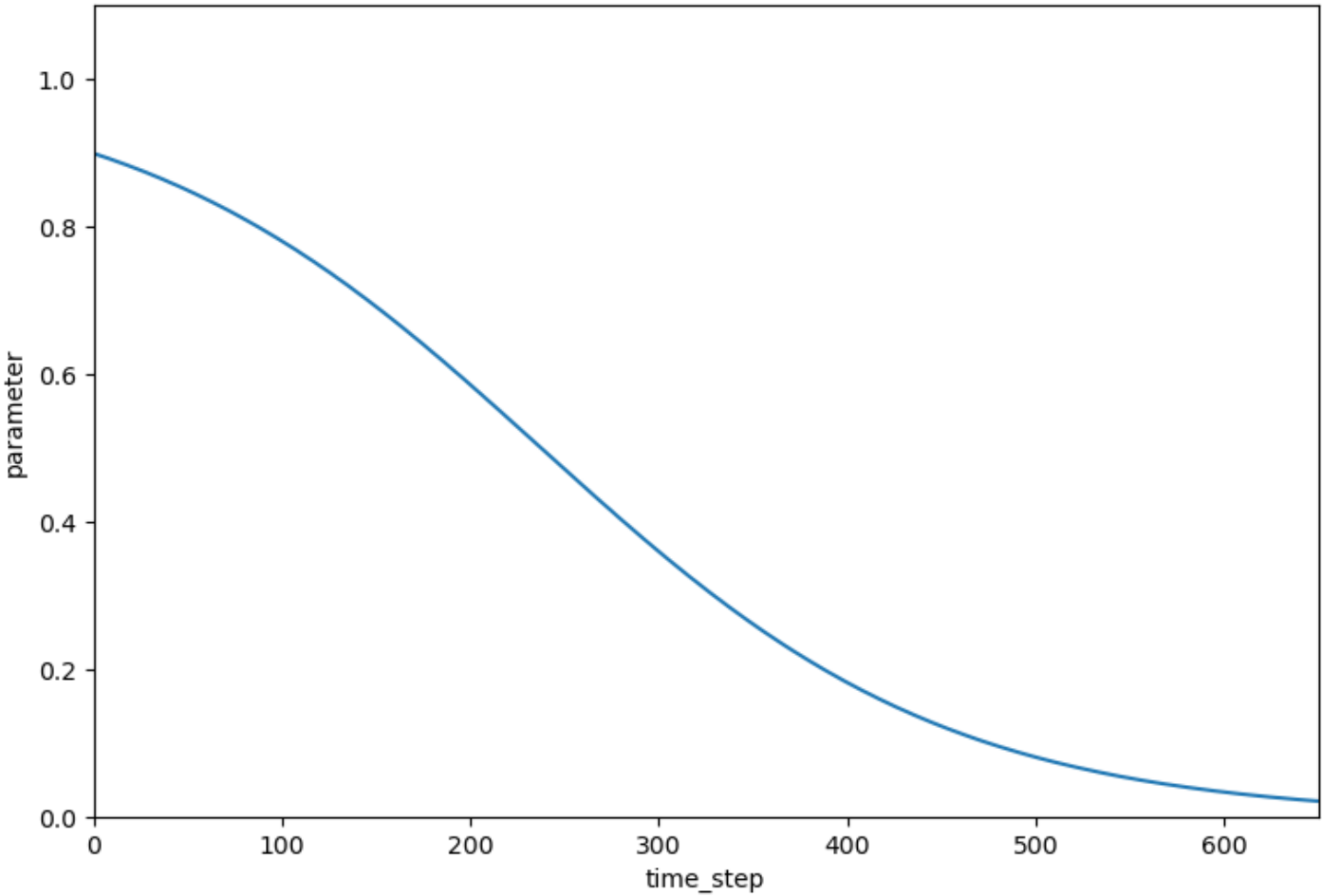
The slider sets the initial condition of the variable

Entrée [8]:

```
fig = plt.figure(figsize=(9, 6), dpi=100)
ax = fig.add_subplot(1, 1, 1)
line, = ax.plot(time_step, update_param(init_x, time_step, dt))
x_label = plt.xlabel("time_step")
y_label = plt.ylabel("parameter")
plt.xlim(0, 650)
plt.ylim(0, 1.1)

def update(w = 0.0):
    line.set_ydata(update_param(w , time_step, dt))
    fig.canvas.draw_idle()

interact(update, w=(0, 1, 0.01));
```



```
interactive(children=(FloatSlider(value=0.0, description='w', max=1.0, step=0.01), Output()), _dom
_classes=('w...
```

Equation with positive d_t .

The slider sets the initial condition of the variable

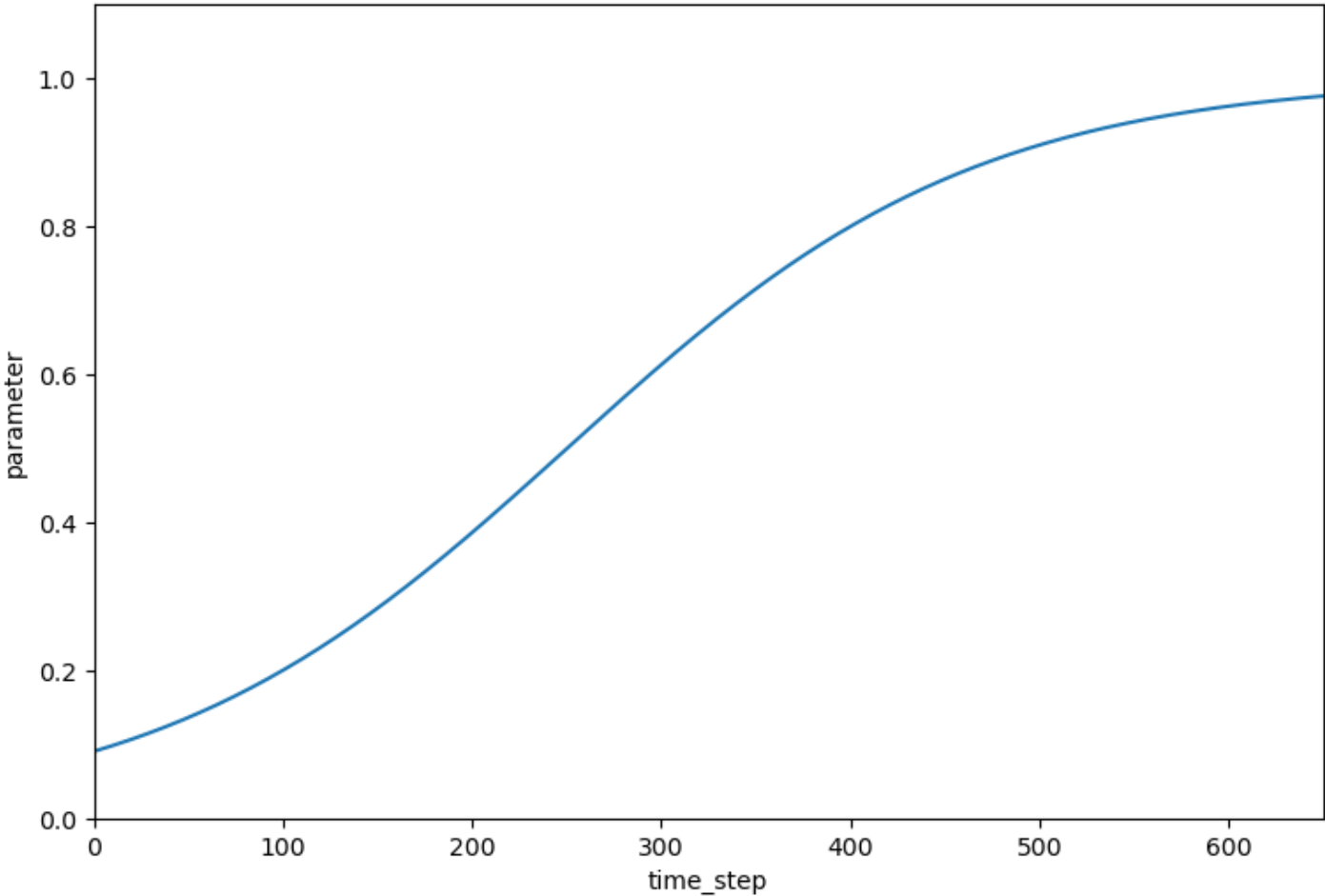
Entrée [9]:

```
init_x = 1
dt = 6
time_step = np.arange(0, 650)

fig = plt.figure(figsize=(9, 6), dpi=100)
ax = fig.add_subplot(1, 1, 1)
line, = ax.plot(time_step, update_param(init_x, time_step, dt))
x_label = plt.xlabel("time_step")
y_label = plt.ylabel("parameter")
plt.xlim(0, 650)
plt.ylim(0, 1.1)

def update(w = 0.0):
    line.set_ydata(update_param(w , time_step, dt))
    fig.canvas.draw_idle()

interact(update, w=(0, 1, 0.01));
```



```
interactive(children=(FloatSlider(value=0.0, description='w', max=1.0, step=0.01), Output()), _dom
_classes=('w...
```

As it can be verified, using the slider in the previous graph, both the curves have two singularities, the first one when $X_{t+1} = 1$ and the second one when $X_{t+1} = 0$. It is possible to verify this by setting the slider on 0 or 1. In both conditions, the curve is saturated, which means that the variable has reached its asymptotic solution.

To solve this problem, we added a ϵ term, where $\epsilon \ll 1$, that avoids saturation.

```
Entrée [10]: x = 0.99
dt = -6
time_step = np.arange(1, 650)

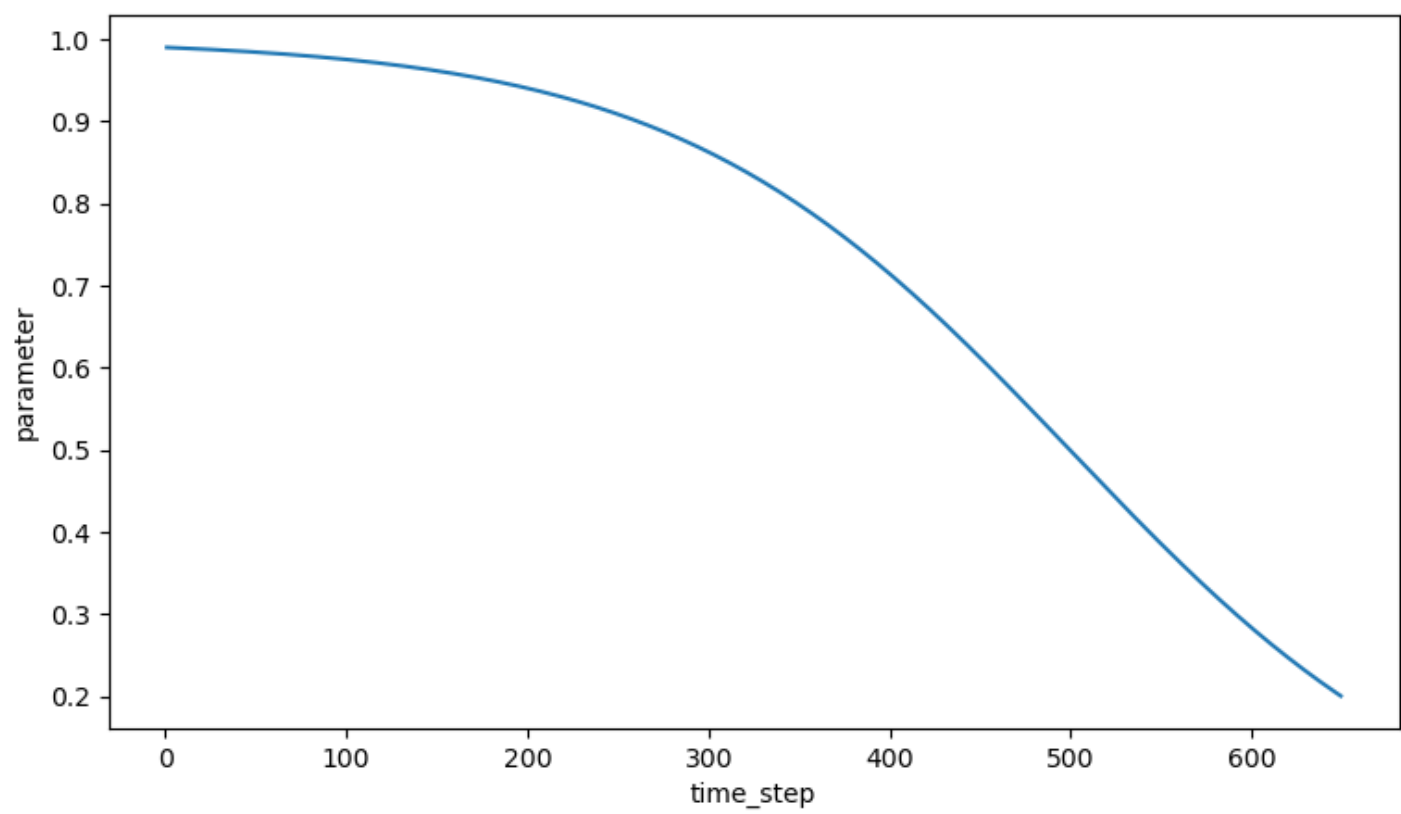
values = []

for i in time_step:
    x += x*(1-x)*(dt/650)
    values.append(x)

fig = plt.figure(figsize = (9, 5))

x_label = plt.xlabel("time_step")
y_label = plt.ylabel("parameter")

plt.plot(time_step, values)
```



Out[10]: [<matplotlib.lines.Line2D at 0x7f5a6ec41100>]

Entrée [11]:

```
x = 0.01

dt = 6

time_step = np.arange(1, 650)

values = []

for i in time_step:

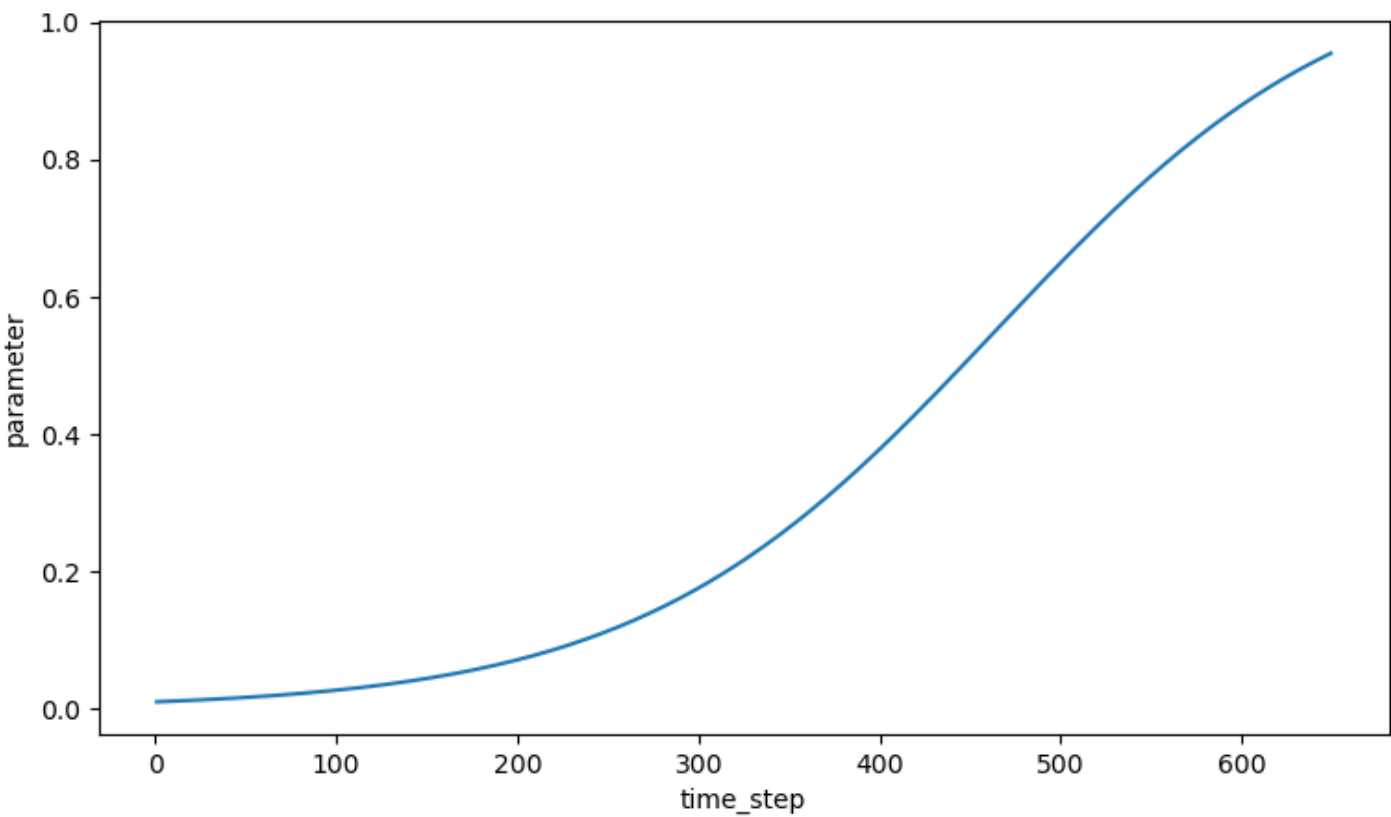
    x += x*(1.1-x)*(dt/650)

    values.append(x)

fig = plt.figure(figsize = (9, 5))

x_label = plt.xlabel("time_step")
y_label = plt.ylabel("parameter")

plt.plot(time_step, values)
```



Out[11]: [

Conclusion

The same scheme is applied to each of the model's variables described previously. The definition of these variables will be improved in the future and adapted to experimental observations. In the future will be also possible to use a different function to describe the evolution of each variable, such as different Hill or Heaviside functions.

Entrée []: