ABM Hands-on Tutorial

Niklas Wulkow and Luzie Helfmann

1st October 2019

Task 1: Movement rule

To start, run `Jupyter notebook frames 1-9` to initialize all methods and parameters.

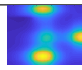
(a) Let $N = 10$ agents move in a suitability landscape $V(x)$ for $T = 200$ time steps. For that, we declare the initial conditions `(Jupyter notebook frame 7)`, set the parameters `(notebook frames 8-9)`, and then iterate over all time steps and over all agents `(frame 10)`.

(b) At each time step, an agent i advances its position according to the discretization of the spatial movement rule. The SDE with $\sigma = 0.02$ is discretized using the Euler-Maruyama scheme with time step $\Delta t = 0.01$ (method `emstep`):

$$X_i(t + \Delta t) = X_i(t) - \nabla V(X_i(t))\Delta t + \sigma\sqrt{\Delta t}\ \eta$$

where $\eta \sim \mathcal{N}(0, 1)$ `(frame 10)`.

(c) Try out different 2D suitability landscapes `(frame 8 and see frame 1 for details)` and plot a movie of the agents' movement on the landscape `(frame 11)`

Potential $V(x, y)$	Gradient ∇V		Name in code
$V(x, y) = 10(x^2 - 1)^{\frac{1}{20}} + y^2$	$\nabla V = (x(x^2 - 1)^{-\frac{19}{20}}, 2y)^T$		doublewell
$V(x, y) = x$	$\nabla V = (1, 0)^T$		linear
$V(x, y) = 0$	$\nabla V = (0, 0)^T$		constant
$V(x, y) =$ landmap image	$\nabla V =$ num. gradient of image		landmap

Task 2: Attraction-repulsion forces between agents

(a) Set the suitability landscape to `constant` and add the attraction-repulsion potential (methods `repulsionattraction` and `repulsionattractiongradient`) to the Euler-Maruyama step

$$X_i(t+\Delta t) = X_i(t) - (\nabla V(X_i(t)) + \nabla U_i(X(t))\,\Delta t + \sigma\sqrt{\Delta t}\,\eta$$

`(frames 8-9 and see frame 1 for details on repulsion-attraction).`
Which effect do you observe?

(b) Tune the parameters (e.g. `distequilibrium` in the method `repulsionattraction`), how does it affect the movement `(frames 6-9)`?

Task 3: Innovation spreading between agents

Discretization of the interaction: an agent changes its state from 0 to 1 within a Δt-sized time interval with probability $p_i(t) = 1 - \exp(-\lambda_i(t)\Delta t)$, where$\lambda_i(t)$ is the infection rate of agent i at time t which is determined by the rate factor γ times the number of infected agents within a radius r (`radius`) of agent i.

(a) Change the code accordingly to include the interaction: using method `determineInteractionRate` to determine the interaction rate of an agent, and using `infectAtRandom` to decide whether to adopt the innovation at random during one time step. `(frame 10)`

(b) Make up your own interaction rule, e.g. what if there is a rate to forget the innovation (i.e. a neighbour-independent rate of an agent to change from state 1 back to 0)? What if there are three states $I_i(t) \in \{0, 1, 2\}$ (SIR model)? `(frames 1 and 10)`