ISSAI

NAZARBAYEV UNIVERSITY | Institute of Smart Systems and Artificial Intelligence

A Particle-based COVID-19 SEIR Epidemic Simulator

Aknur Karabay (aknur.karabay@nu.edu.kz),
Askat Kuzdeuov, Madina Abdrakhmanova and Huseyin Atakan Varol.

25 June 2021, Friday

www.issai.nu.edu.kz

The particle Model

In our simplified model, an individual is modelled as particle $\boldsymbol{p}$ with the following parameters:

$$\boldsymbol{p} = [\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{e}, \boldsymbol{t}, \boldsymbol{ag}]$$

where $\boldsymbol{x}$ – position of the particle on the 2D map,

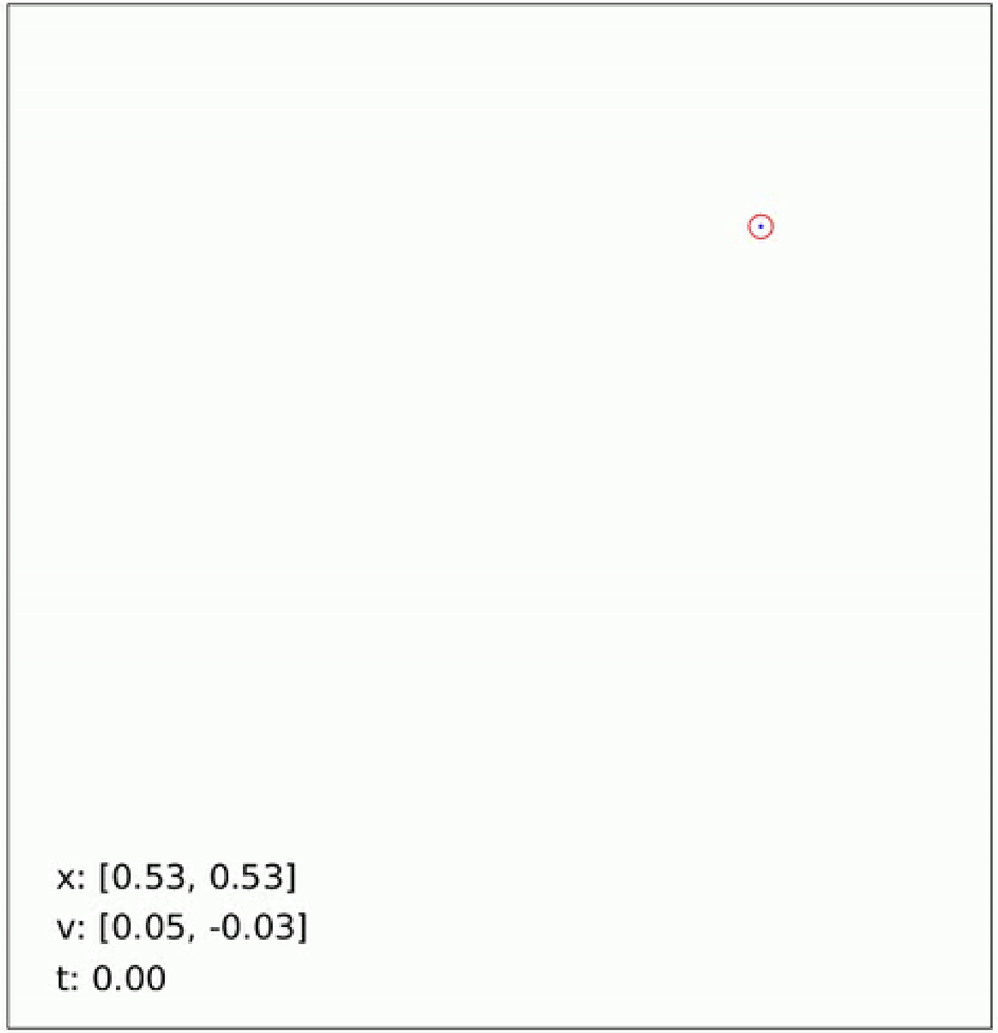
$\boldsymbol{v}$ – the particle velocity,

$\boldsymbol{e}$ – the epidemic state of the particle

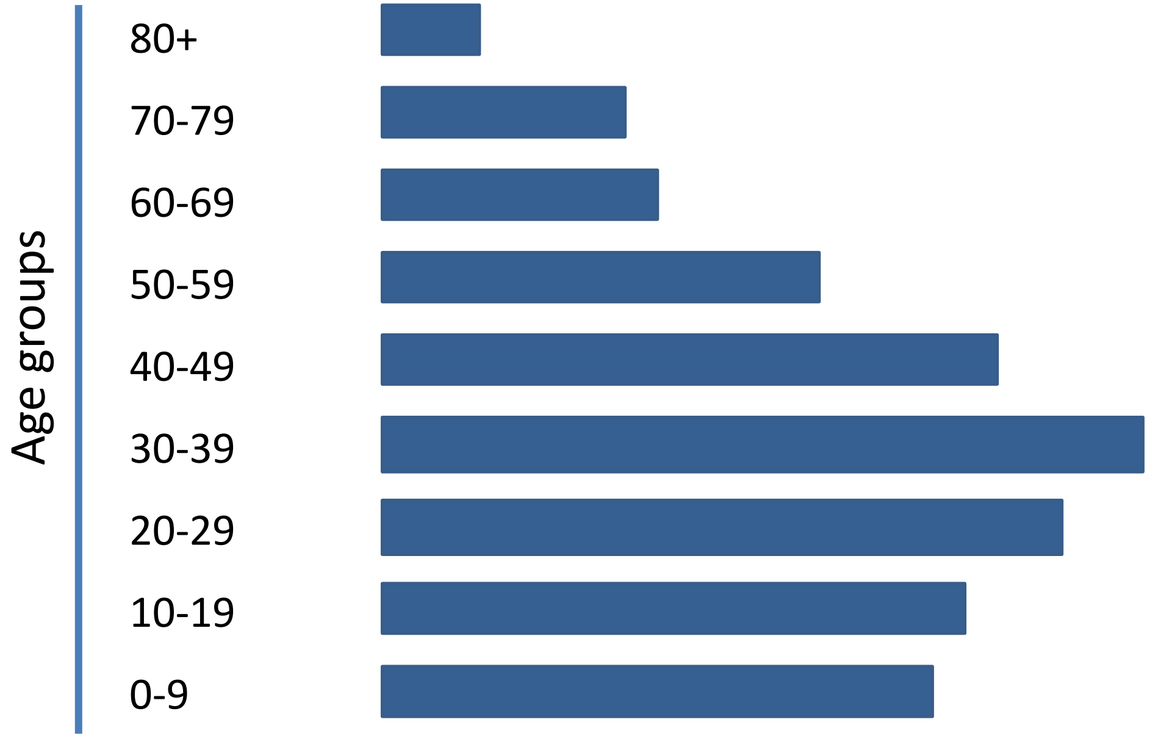
(susceptible (0), exposed (1), infected (2), severe infected (7), recovered (3), dead (4)),

$\boldsymbol{t}$ – the time of the particle in the current epidemic state,

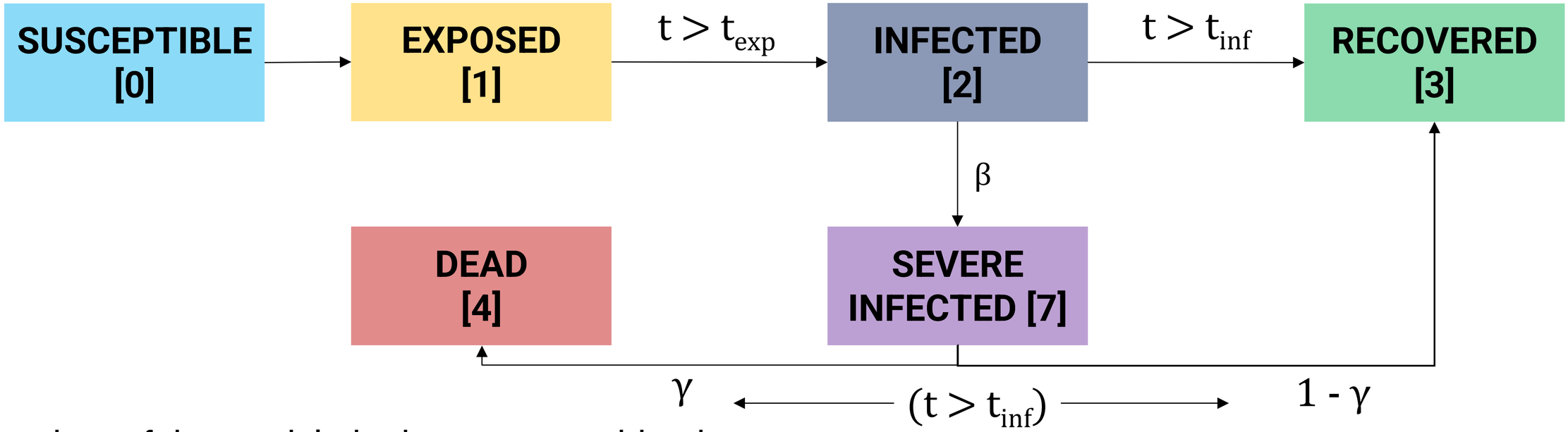
$\boldsymbol{ag}$ – the age group



The age pyramid



The statechart of the Particle-based SEIR simulator.



t – the time of the particle in the current epidemic state

t_{exp} – disease exposure period (2 days)

t_{inf} – disease infection period (5 days)

β – rate of infected particles transitioning to the severe infected state based on the age group

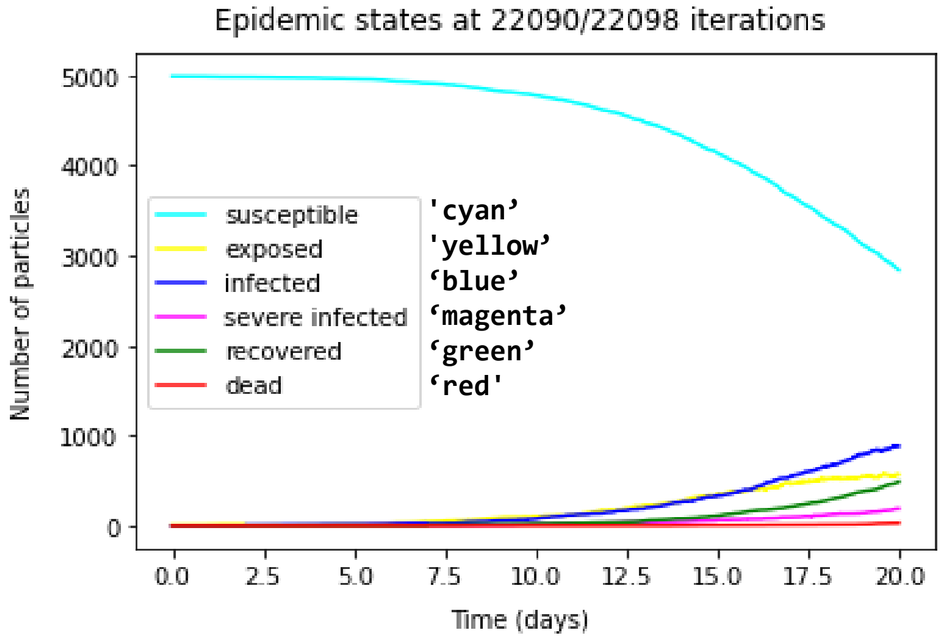
γ – mortality rate

Task 1: Write a function of `Class Particles` called `update_coordinates`. The function takes `simulator,` an object of `Class Simulator.` At each iteration update the coordinate by the distance moved at the current iteration. “(Hint: `simulator.delta_t` is the time a particle spends in each iteration.)”

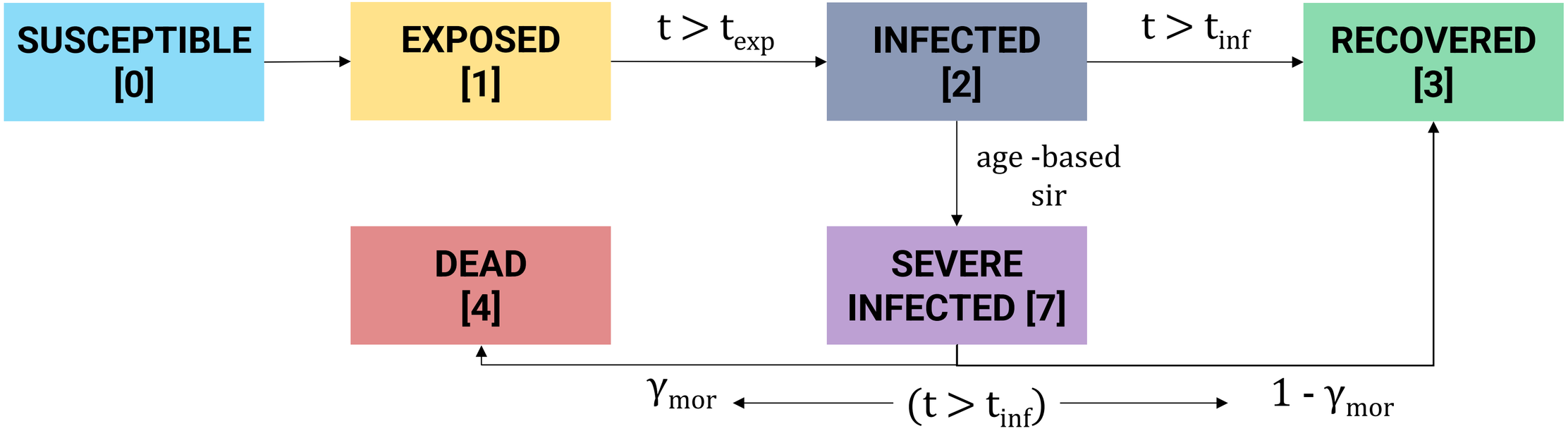
Note, the particles must stay inside of the 2D boundaries, set to [-1,1] for both dimensions. If a particle reaches one of the borders, it should be sent to the opposite side. For example, if x > 1, then update to x = -1.

Task 2: Write a function of `Class Particles` called `plot` that visualizes the epidemic curves for each state. The function takes two parameters: 1) `simulator,` an object of class Simulator; 2) `i`, the id number of the current iteration. The id should be featured in the title of the plot. The function should save the plot in .png format in the `plots` subdirectory under the name `states_i.png`, where `i` is the id number.

To test the function, you are provided with `data_for_plots.p`. See test # 2 in the main method of `particles.py.` The example below is the result of that test.



Task 3: Write a function of `Class Simulator` called `susceptible_to_exposed` that updates the epidemic status of particles from susceptible to exposed. The function takes two parameters: 1) `model`, an object of `Class Particles`; 2) `susceptible_contacted`, a list of indices of susceptible particles that were close to contagious particles (exposed, infected, severe infected) at the current iteration. Using these indices the function should: 1) update the corresponding elements in the `model.epidemic_state` array to the exposed state; 2) reset the corresponding elements in the `model.time_cur_state` array to 0.



Task 4: Write a function of `Class Simulator` called `infected_to_recovered` following the example code provided for the `exposed_to_infected` method. The function takes `model,` an object of `Class Particles.` The function should: 1) get the indices of the particles whose time at the current state have reached `T_INF` in `model.time_cur_state` array; 2) for these indices update the `model.epidemic_state` to recovered and `model.time_cur_state` to 0.