Reflection for SIR model simulation

In this computational research, I meanly encountered with 2 problems. The first one is how to output the animation, or in other words, the gif graphics, in the jupyter notebook; and the other one is how to ensure the output can be run smoothly.

The first one is actually about how to set the output graphics token by a line of code. Different from other IDEs, jupyter notebook has nowhere to set the output token manually so that the compiler can realize the function of outputting a gif graph by adding some code automatically. As I get experience of using magic command in notebook to realize some specific functions, including outputting interactable graphics, I use built-in-function, help, to search the magic code which can output gif graphics, and then successfully found it. This difficulty strengthened by ability of using help function the get the answer of some minor but key problems that is really difficult to find on the internet or public programming societies.

The second problem is how to make the animation running smoothly. As I used the code I programmed the simulate the elastic collision between particles last semester to simulate the moving people in a closed environment, the number of calculation of one loop increases exponentially with the increase of the number of people simulated. In the given example of pandemic spread, there are 1000 person, but my simulation could only support a simulation of 200 particles to run in a smooth way, and for 1000 particles, the frame per second will decrease to 1 or 2. As a result, a total number of 100 particles is chosen to simulate how the pandemic is spread among people. But this brings another problem, that is, every particle represents 10 people, and I should find a way to indicate group of people infected, recovered or susceptible less than 10 reasonable. A reasonable method is easy to come up, using mixed color, but this need me to achieve some functions that aren't necessary if I choose 1000 particles precisely, such as calculate percentage of color, locating range of particles that include a certain kind of people.

During this research, I have discovered that using numerical integration algorithm to tackle questions involving solving differential equations and visualize its time evolution process is really helpful to the understanding of the question.