PROJECT 4 DESIGN DOCUMENT

**Introduction**

This project develops a simulation model for the spread of a virus within a limited area and with a predefined number of people. The area is divided into regions and people move from one region to another in random directions.

Useful variables to better understand the implementation:

* N: total number of individuals.
* P: number of processes.
* R: number of regions:
* Ax: length of the Area.
* Ay: width of the Area.
* Rx: length of each region.
* Ry: width of each region.
* t: time after which the position of individuals is recomputed.
* d: spreading distance.
* v: moving speed of each individual.

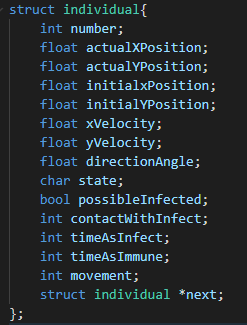
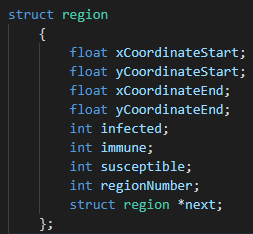
Some of the individuals are already infected and can infect others. If a susceptible individual stays close for more than 10 minutes to an infected individual it becomes infected. After 10 days of infection an individual becomes immune. After 3 months of immunity an individual returns susceptible.

At the end of each simulated day the number of individuals for each region is shown and the health state of those individuals.

**Implementation**

Given the fact that the computation is performed by P different processes, the first useful computation is to divide the individuals and regions among the different processes. Each process updates the position of the individuals it deals with and the number of individuals in the regions it deals with.

Where n is the number of individuals for each process and r the number of regions for each process. The eventual rest of the division is given to the last process. Individuals and region are saved as structs with all useful information in two different linked lists.



After this initial computation, every process creates a matrix M (P x n) in which the rows represent the processes, and the columns are the individuals.

So, the first thing that each process does after creating the matrix is updating the row corresponding to itself (process y uses row y of the matrix).

Once this operation is completed, the main cycle that marks the operation of each process begins. Since the time t represents the time that passes in the simulation after which the position of the individuals must be updated, a single cycle represents that time t of simulation.

The operations that each process does in a cycle are:

* Checking if a day has passed. Since one cycle is t seconds, a day is a number of cycles equal to . If a day has passed each process prints the state of each individual in each region.
* For each individual:
  + Check if there’s a change in the condition of the individual. This can happen if one of the conditions depicted in the introduction is met.
  + Compute the updated position of the individual. The movement is a simple uniform motion in one random direction. So at each time the new position is. . Where vx and vy are simply the velocity of the individual in the corresponding direction. If, in the computation, an individual is found outside the area, his position is recomputed with a new random direction in the opposite part of the area.
  + Update the individual’s matrix M.
* Send its row of the matrix as an MPI message to all other processes.
* Receive their row form the other processes. At the end of this operation each process has a complete matrix of all the individuals that are in the area.
* Using matrix M each process does two actions:
  + Check if its susceptible individuals are near an infected one.
  + Update the individuals in each region.
* Before starting the next cycle there is an MPI Barrier, to make sure that each process starts each cycle in the same moment.