# Parallel Design Patterns Coursework2 Report

B156924

April 10, 2020

# Contents

# 1 Introduction

Simulation calculations can help scientists take advantage of the powerful computing capabilities of supercomputers to analyze and predict complex problems in the real world. This technology has applied to many scientific field, such as infectology, biology, physics and astronomy. In the past few months, the COVID-19 has been spreading among many countries. A lot of data scientists have simulated the spread of the virus and predict the trends of infection and death numbers, which provides useful information to government's actions. At the same time, many scholars have demonstrated to the public the positive effects of home isolation on mitigating the epidemic through scientific simulation.

Different simulations are applicable to different program structures and patterns. For the case in this coursework, it is to help biologists to simulate the spread of disease through the squirrel population for discovering information about protecting the animals. As mentioned in the previous report, the Actor algorithm pattern is the most suitable one among all patterns since it can easily simulate the behavior of individual squirrels in the natural environment and the interaction between squirrels and the environment.

Combined with Master/Worker implementation strategy, the model can be easily constructed. MPI(Message-passing Interface) is applied for message passing between actors. This model is written in C and can be compiled with GCC 4.8.5. The code is tested on 7 computing nodes(each node has 36 cores) on the Cirrus platform with Linux system.

This report is organized as follows. In chapter 2 the design of this simulation program is illustrated. Chapter 3 presents the results of the simulation including correctness test and output analysis. Chapter 4 presents the performance analysis for this program. Finally, in chapter 5, the overall performance of this program is discussed, followed with ideas about further improvement.

# 2 Program design

In this chapter, the design of this program is described. The Master/Worker strategy is implemented to set a master process to allocate, manage and kill other worker processes. The provided file *pool.c* makes it easy to achieve this strategy. For the Actor pattern, there are 4 actor in this program: *Master, Control, Cell* and *Squirrel*. The following sections explain the details of their work.

## 2.1 Master/Worker strategy

Master/Worker strategy is used for tasks that are not tightly coupled. The master process can allocate processes for the coming work to execute and remind them to stop when the program is going to terminate. The processes can be reused when the previous work is done, so this is conducive to the workload balance.

The file *pool.c* provides a dynamic process pool, and there are a bunch of functions designed for this pool. For this simulation, the master process is used for initializing *processPoolInit()* and shutting down *processPoolFinalise()* the pool, and supervising *masterPool()* each process's work status. For each worker process, it is able to request a new process *startWorkerProcess()* from the master, end its work to sleep *workerSleep()* and request to shut down *shutdownPool()* the pool.

By calling these functions, the simulation program can achieve process allocation and management.

## 2.2 Actor pattern

In the Actor pattern, actors can perform their own computational work when messages haven't arrived. They are also able to create a new actor, die and send messages to others. These properties are suitable for the biological model.

For this simulation, there are four actors designed: *Master, Control, Cell* and *Squirrel*. Figure 1 is the interaction diagram of these actors. It is worth noting that the master actor only allocates a process for the control actor, and the control actor allocates the cell actors and squirrel actors including initialized healthy and infected squirrels. The control actor also has a clock function, it will send messages to cell actors when a new month begins and receive cells' updated information. The squirrel actors don't know the clock, and they spend their lives on interacting with the cell actors and the control actor and judging the healthy and reproduce conditions.

In the next sections, the work for each type of actor is explained.

### 2.2.1 Master actor

In the main function, by calling the *processPoolInit()* function, the master process is created with its status code equaling 2. The work for the master actor is very simple. The first thing is to allocate a process to the control actor, the other thing is to continually check the status of each actor. If some actor calls *shutdownPool()* to request the master to terminate the program, the master will break the check loop and execute *processPoolFinalise()* function, which will send each actors messages to stop their works and to sleep.

If the status code is 1, it means all worker processes. The code in this part is executed in every worker process. There is a while loop designed here keeping receiving messages from other actors who have requested a new process. The message content is the type of actors corresponding to different work: CONTROL_ACTOR, CELL_ACTOR, SQUIRREL_ACTOR, NEW_SQUIRREL_ACTOR and INFECTED_SQUIRREL_ACTOR.

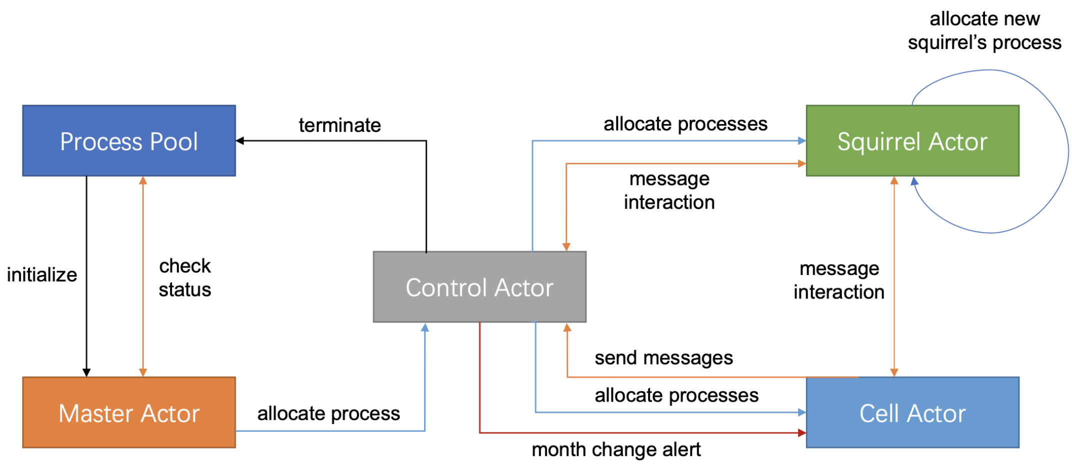


Figure 1: This is the interaction diagram of the actors in the simulation. The process pool is the combination of a bunch of functions that support Master/Worker structure. Therefore, we can regard it as a black box. In the beginning, the master actor is initialized by the process pool. The master actor allocates a new process as the control actor, and then it will continually check the changes of other actors' statuses. As for the control actor, it allocates the processes to all cell actors and squirrel actors. The control actor needs to alert all cell actors per month to update their information and send it to the control actor. The squirrel actor always interacts with cell actors and the control actor and sometimes can born a new squirrel and allocate its process.

When an actor wants to request for a new process, it needs to call *startWorkerProcess()* first and send the type of this actor to its process rank. In this way, a new actor is set up.

#### 2.2.2 Control actor

The control actor is mainly responsible for allocating initialized cell actors and healthy and infected squirrel actors, playing as a clock to interact with cell actors when the month changes and recording the number of alive, infected and dead squirrels for each month. Figure 2 is the workflow of the control actor.

For initializing actors, as mentioned before, the control actor can allocate a process for a new actor and send the work type to this process.

The next step is being a clock and a recorder. when a month passes, the control actor will send messages to each cell actor to remind them of updating their populationInflux and infectionLevel and then these cells are required to send back their information to the control actor. All cells' information is received with a designed MPI_Type_vector and stored in a 2-dimensional array in order.

During each month, the control actor keeps receiving messages from squirrels though MPI_Iprobe() detection. There are three types of messages: a new squirrel is born, a healthy squirrel is infected, and an infected squirrel has died. The control actor records the information and displays this information when the month changes.

It is worth noting that when a new month starts, the control actor will judge whether the current squirrel number is 0 or over 200. If all squirrels die, the program needs to be terminated. The control actor will break the loop and call *shutdownPool()* . If the squirrel number is over 200, it means the program may have bugs, therefore, an assert function is called to terminate the program straightly.

When the overall simulation time has passed, the control actor will call *shutdownPool()* to tell the master to terminate the program.

#### 2.2.3 Cell actor

The cell actor's work is to record the visits of healthy and infected squirrels, compute populationInflux and infectionLevel per month and send the updated information to the control actor per month. To store a single month's populationInflux and infectionLevel, two int arrays *month_populationInflux*, *month_infectionlevel* are created with the size of model months. Figure 3 is the workflow of a cell actor.

When the simulation starts, the cell actor waits for the control actor's first-month message to start working. In this way, the cells won't interact with squirrels immediately after being created, and the accuracy of time can be guaranteed.

Once it receives the month signal, it will jump into a loop. For each iteration, the cell will check whether it needs to terminate. Then a MPI_Iprobe() will detect the coming

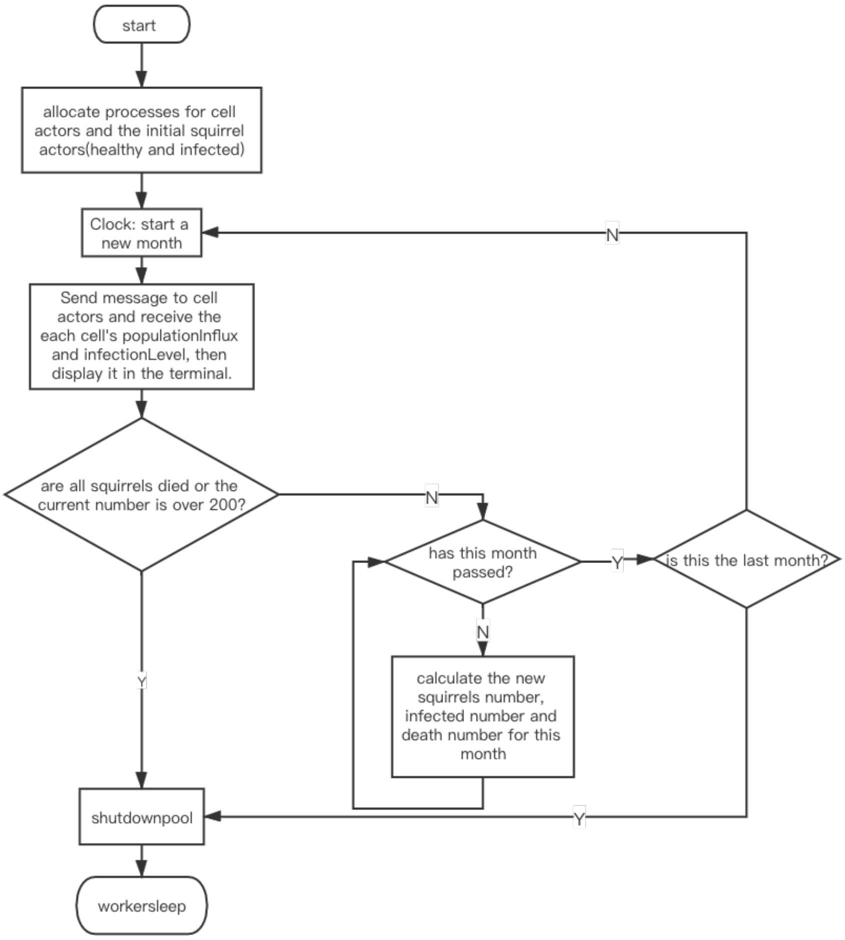


Figure 2: This is the workflow of the control actor.

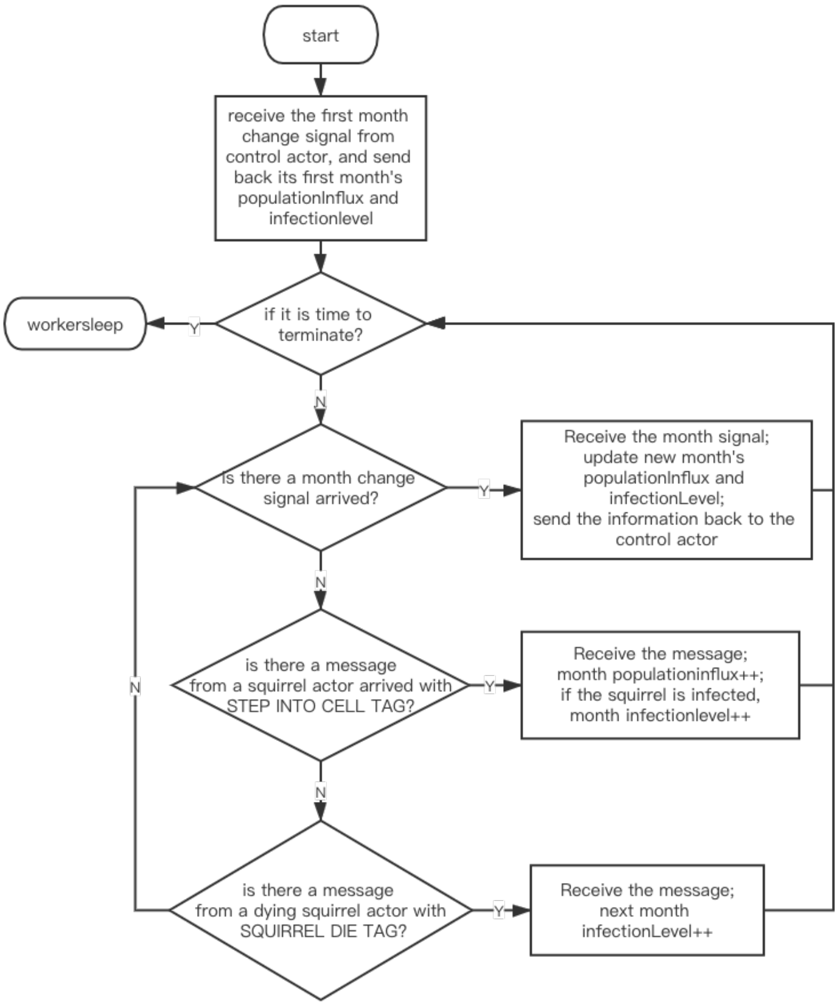


Figure 3: This is the workflow of a cell actor.

messages. If its tag is 'MONTH_ALERT_TAG', the cell needs to update its information and send it back to the control. If it is a 'STEP_INTO_CELL_TAG', it means a squirrel has stepped into this cell. The cell will record this visit to its *month_populationInflux*. If the squirrel is infected, *month_infectionlevel* will add 1. Finally, if its tag is 'SQUIRREL_DIE_TAG', it means that a squirrel is about to die in this cell, and the cell will update its next month's infectionLevel in advance since the virus will remain for 2 months after the squirrel dies.

The cell will keep working in this loop until it is asked to terminate.

### 2.2.4 Squirrel actor

The squirrel actor simulates the real squirrel's behavior in nature. It will keep jumping between cells and can possibly born a new squirrel, get infected and die. Figure 4 is the workflow of a squirrel actor.

Since the squirrel sends many messages to the control actor and cell actors, the deadlock can easily appear here when the program is terminated while the squirrel is sending a message. Therefore, before each send, the squirrel will call *workerSleep()* to check whether it is asked to stop working. Besides, the send buffer is allocated with a big size to ensure that the squirrel won't run out of buffer when using the Bsend().

When the squirrel steps into a cell, it will send its information to the cell and receive the cell's populationInflux and infectionLevel. In order to store the cell information for each step properly, a queue is designed for the squirrel's populationInflux trajectory and infectionLevel trajectory. The size of the queue is 50. When the queue is full, the earliest step's data will be popped and the newest data will be pushed into the queue. Besides, the mean of all data can be calculated easily and all elements can be printed. Such a design can avoid the wasted storage of useless data and more convenient to calculate.

After storing the cell's data, the squirrel is supposed to test whether it is infected. If the squirrel is infected, the *infected_step* int variable will keep recording its steps since the first step when it is tested infected. If the *infected_step* is more than 50, the squirrel will have 1/6 possibility to death for each next step. When the squirrel is tested dead in a cell, the dying squirrel will send a message with SQUIRREL_DIE_TAG to the corresponding cell and a message to the control actor to record its death. After that, it will break the loop and call *workerSleep()* to finish its life.

The final test is whether the squirrel will bear a baby in this step. If it is true, firstly the squirrel needs to ask the control actor whether the current squirrels are more than 200. If not, the squirrel is allowed to allocate a new process for the baby and deliver its position to its baby.

It is worth mentioning that a *squirrel_stall()* function is designed for adjusting the moving speed of squirrels. When the supercomputer computes quickly, all squirrels may die in 2-3 months, the trends of born and death can hardly be analyzed. Therefore, this function can solve this problem.

# 3 Results correctness

The output of the simulation can be displayed in the terminal. For each month, the alive number, infected number, healthy number, last month death and born number, and the populationInflux and infectionLevel for each cell are presented with good format. With the same parameters, the results vary in every execution but the general trends are similar. Therefore, the correctness of this simulation needs to be tested to guarantee its accuracy. There are two ways to test its correctness.

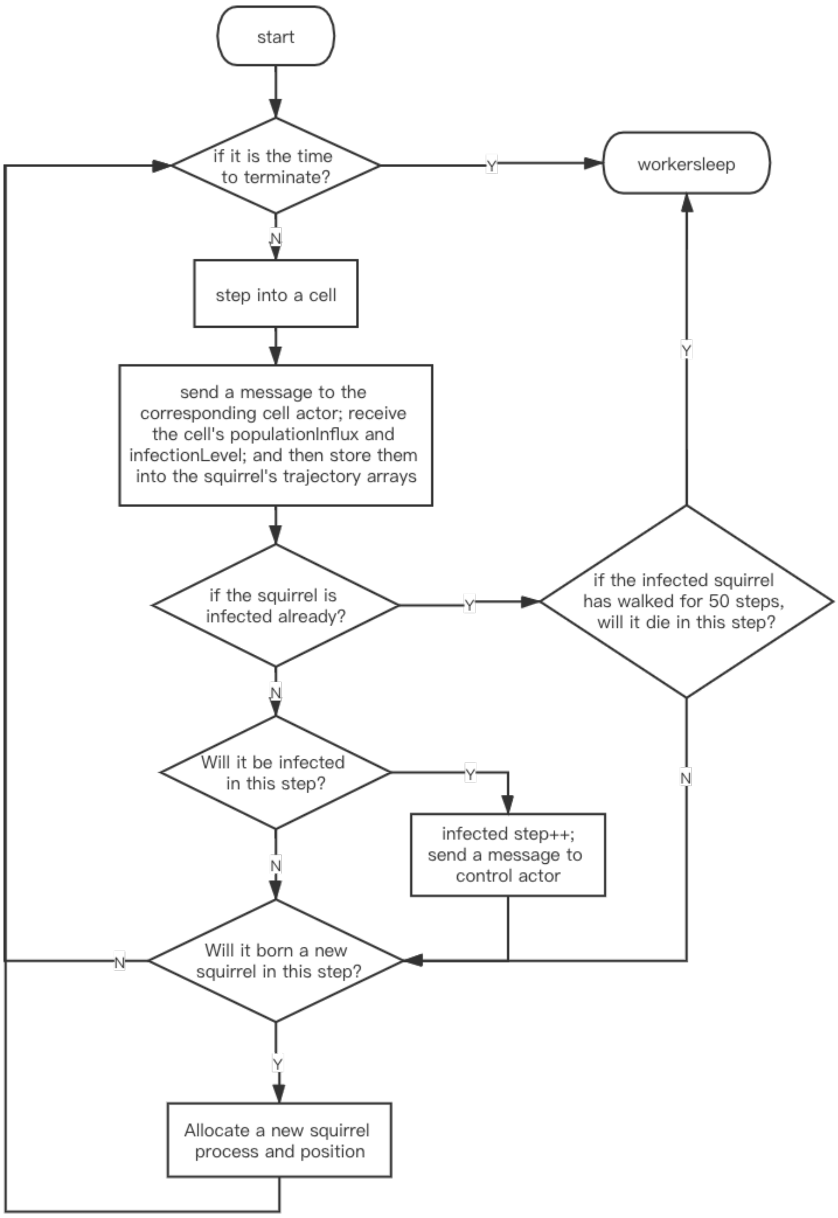


Figure 4: This is the workflow of a squirrel actor.

```
SIMULATION STARTS.
----------------------------------------------------------------------------------------------------------------------------
MONTH: 1        ALIVE: 38       INFECTED: 4     HEALTHY: 34     LAST MONTH DEATH: 0     LAST MONTH BORN: 0
----------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
InfectionLevel          0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
----------------------------------------------------------------------------------------------------------------------------
MONTH: 2        ALIVE: 38       INFECTED: 4     HEALTHY: 34     LAST MONTH DEATH: 0     LAST MONTH BORN: 0
----------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        107     102     113     100     100     82      104     104     109     93      95      113     106     97      105     90
InfectionLevel          14      10      14      13      11      11      7       7       11      7       11      14      9       6       7       4
----------------------------------------------------------------------------------------------------------------------------
MONTH: 3        ALIVE: 34       INFECTED: 7     HEALTHY: 27     LAST MONTH DEATH: 5     LAST MONTH BORN: 1
----------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        207     188     211     202     202     183     217     206     245     211     205     225     252     196     211     185
InfectionLevel          22      20      27      30      22      27      24      20      28      16      27      31      27      14      17      11
----------------------------------------------------------------------------------------------------------------------------
MONTH: 4        ALIVE: 37       INFECTED: 18    HEALTHY: 19     LAST MONTH DEATH: 3     LAST MONTH BORN: 6
----------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        299     283     320     300     313     302     323     317     366     316     318     331     348     309     309     280
InfectionLevel          49      49      55      49      62      67      61      60      68      54      58      63      60      62      52      50
----------------------------------------------------------------------------------------------------------------------------
MONTH: 5        ALIVE: 23       INFECTED: 13    HEALTHY: 10     LAST MONTH DEATH: 17    LAST MONTH BORN: 3
----------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        273     254     260     285     310     297     294     301     336     320     299     292     336     287     277     278
InfectionLevel          86      82      75      84      97      91      84      98      95      100     80      84      93      98      86      91
----------------------------------------------------------------------------------------------------------------------------
MONTH: 6        ALIVE: 19       INFECTED: 16    HEALTHY: 3      LAST MONTH DEATH: 8     LAST MONTH BORN: 4
----------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        216     215     221     234     259     246     233     255     266     243     245     239     233     236     216     225
InfectionLevel          78      75      78      88      80      77      81      91      98      89      80      86      83      83      78      84
----------------------------------------------------------------------------------------------------------------------------
MONTH: 7        ALIVE: 7        INFECTED: 7     HEALTHY: 0      LAST MONTH DEATH: 13    LAST MONTH BORN: 1
----------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        153     154     145     167     182     151     159     170     167     173     164     162     162     154     149     153
InfectionLevel          59      63      77      64      62      60      72      62      76      65      74      72      55      66      64      57
----------------------------------------------------------------------------------------------------------------------------
MONTH: 8        ALIVE: 3        INFECTED: 3     HEALTHY: 0      LAST MONTH DEATH: 5     LAST MONTH BORN: 1
----------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        85      90      103     91      93      88      90      88      102     87      97      98      73      86      81      79
InfectionLevel          39      42      43      36      36      37      37      28      36      44      42      35      30      36      36      36
----------------------------------------------------------------------------------------------------------------------------
MONTH: 9        ALIVE: 1        INFECTED: 1     HEALTHY: 0      LAST MONTH DEATH: 3     LAST MONTH BORN: 1
----------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        49      49      46      47      49      41      47      37      46      51      46      44      36      44      41      41
InfectionLevel          18      16      13      16      14      17      13      10      20      15      16      15      11      15      10      17
----------------------------------------------------------------------------------------------------------------------------
MONTH: 10       ALIVE: 0        INFECTED: 0     HEALTHY: 0      LAST MONTH DEATH: 1     LAST MONTH BORN: 0
----------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        21      18      14      17      18      17      17      13      26      19      15      17      12      15      13      20
InfectionLevel          7       9       3       9       9       3       10      6       8       5       7       6       5       8       7       4
----------------------------------------------------------------------------------------------------------------------------
ALL SQUIRRELS DIED.
SIMULATION ENDS.
```

Figure 5: The output of a simulation. The parameters are: 34 healthy squirrels, 4 infected squirrels, 16 cell. The birth probability is 300.0 and the catch disease probability is 500.0.

## 3.1 Method 1

The first method is to look at the ALIVE, LAST MONTH DEATH, and LAST MONTH BORN presented in the output(Figure 5). If all squirrels have died before the terminal, these values should conform to the following formula(1):

$$ALIVE_0 + \sum LAST\_MONTH\_BORN = \sum LAST\_MONTH\_DEATH \quad (1)$$

The ALIVE_0 means the first month's ALIVE value-the initial squirrel number(including healthy and infected squirrels). The output of this simulation can pass this test.

```
SIMULATION STARTS.
--------------------------------------------------------------------------------------------------------------------------------
MONTH: 1        ALIVE: 38       INFECTED: 4     HEALTHY: 34     LAST MONTH DEATH: 0     LAST MONTH BORN: 0
--------------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
InfectionLevel          0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
--------------------------------------------------------------------------------------------------------------------------------
cell 0 month 1 month_populationInflux = 134, month_infectionlevel = 16
MONTH: 2        ALIVE: 37       INFECTED: 3     HEALTHY: 34     LAST MONTH DEATH: 1     LAST MONTH BORN: 0
--------------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        134     122     134     125     127     107     115     123     147     127     121     133     142     122     136     113
InfectionLevel          16      13      16      20      15      13      8       8       13      8       13      15      13      9       8       5
--------------------------------------------------------------------------------------------------------------------------------
cell 0 month 2 month_populationInflux = 111, month_infectionlevel = 9
MONTH: 3        ALIVE: 37       INFECTED: 10    HEALTHY: 27     LAST MONTH DEATH: 3     LAST MONTH BORN: 3
--------------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        245     217     257     238     243     220     238     240     279     245     234     251     288     220     242     226
InfectionLevel          25      21      29      26      25      26      20      20      22      17      24      30      29      21      18      12
--------------------------------------------------------------------------------------------------------------------------------
cell 0 month 3 month_populationInflux = 124, month_infectionlevel = 54
MONTH: 4        ALIVE: 36       INFECTED: 15    HEALTHY: 21     LAST MONTH DEATH: 8     LAST MONTH BORN: 7
--------------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        369     326     368     360     376     335     353     379     413     366     372     371     417     350     354     335
InfectionLevel          63      53      56      49      58      52      59      66      66      57      60      57      66      60      52      56
--------------------------------------------------------------------------------------------------------------------------------
cell 0 month 4 month_populationInflux = 115, month_infectionlevel = 70
MONTH: 5        ALIVE: 25       INFECTED: 18    HEALTHY: 7      LAST MONTH DEATH: 14    LAST MONTH BORN: 3
--------------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        350     299     334     348     357     334     341     348     366     332     365     324     358     308     328     317
InfectionLevel          124     99      105     112     106     112     105     114     120     100     117     95      94      98      103     108
--------------------------------------------------------------------------------------------------------------------------------
cell 0 month 5 month_populationInflux = 43, month_infectionlevel = 36
MONTH: 6        ALIVE: 6        INFECTED: 5     HEALTHY: 1      LAST MONTH DEATH: 19    LAST MONTH BORN: 0
--------------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        282     241     265     277     282     276     272     274     285     272     299     246     260     251     287     250
InfectionLevel          106     84      106     103     84      116     104     95      105     97      102     87      81      82      117     96
--------------------------------------------------------------------------------------------------------------------------------
cell 0 month 6 month_populationInflux = 10, month_infectionlevel = 9
MONTH: 7        ALIVE: 1        INFECTED: 1     HEALTHY: 0      LAST MONTH DEATH: 5     LAST MONTH BORN: 0
--------------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        168     142     158     170     159     172     162     148     157     163     171     140     135     133     187     155
InfectionLevel          45      41      49      52      36      53      51      48      50      58      45      49      42      45      67      52
--------------------------------------------------------------------------------------------------------------------------------
cell 0 month 7 month_populationInflux = 1, month_infectionlevel = 1
MONTH: 8        ALIVE: 0        INFECTED: 0     HEALTHY: 0      LAST MONTH DEATH: 1     LAST MONTH BORN: 0
--------------------------------------------------------------------------------------------------------------------------------
CellId                  0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
PopulationInflux        54      48      61      57      51      68      59      57      57      70      57      54      52      55      79      61
InfectionLevel          10      12      8       19      10      13      6       15      8       13      11      15      5       15      14      16
--------------------------------------------------------------------------------------------------------------------------------
ALL SQUIRRELS DIED.
SIMULATION ENDS.
```

Figure 6: The output of a simulation with printed cell information. The parameters are the same as that of Figure 5.

## 3.2 Method 2

The second method is to print a cell's single month's populationInflux and infectionLevel. In this way, we can test that the monthly updated information is correct. Figure 6 is the output with CELL_DEBUG = 1 and the tested cell Id is 0. The monthly update method is to sum up the past-3-month population and the past-2-month infection. From the Figure 6 it is easy to test the calculation of populationInflux and infectionLevel for cell 0. For example, the cell 0 populationInflux in month 4 can be calculated by summing up the cell 0 month_populationInflux of month 1, 2 and 3. In this way, we can prove that the values of monthly updated populationInflux and infectionLevel are convincing in this simulation.

# 4 Conclusions

This program can be successfully executed with different numbers of initial healthy and infected squirrels. It will appear an error when there are not enough processes provided. When the number of the squirrel is larger than 200, the program will appear the error message, and when all squirrels die, the program can be terminated immediately. Besides, the Macro definition 'CN' in file *simulation.h* can help in changing the number of cells.

However, it also has some issues needed to improve. One is that there are no user-define parameters provided and you can only modify parameters in *simulation.h*. Another issue is that there is not a framework constructed for this simulation. Therefore, it is not very portable to apply to other cases.