Monte Carlo simulation of pandemic

Modelling of Physical Systems

Piotr Cenda
Kraków, 29.03.2022

1 Model

The initial code of model of pandemic simulation was provided by the tutor. It included default (similar to real-life, COVID-19 virus properites) parameters, which impact on model was tested during task nr 1. Code was also used as base model during task nr 2, when isolation mechanism was introduced. Code was as the following:

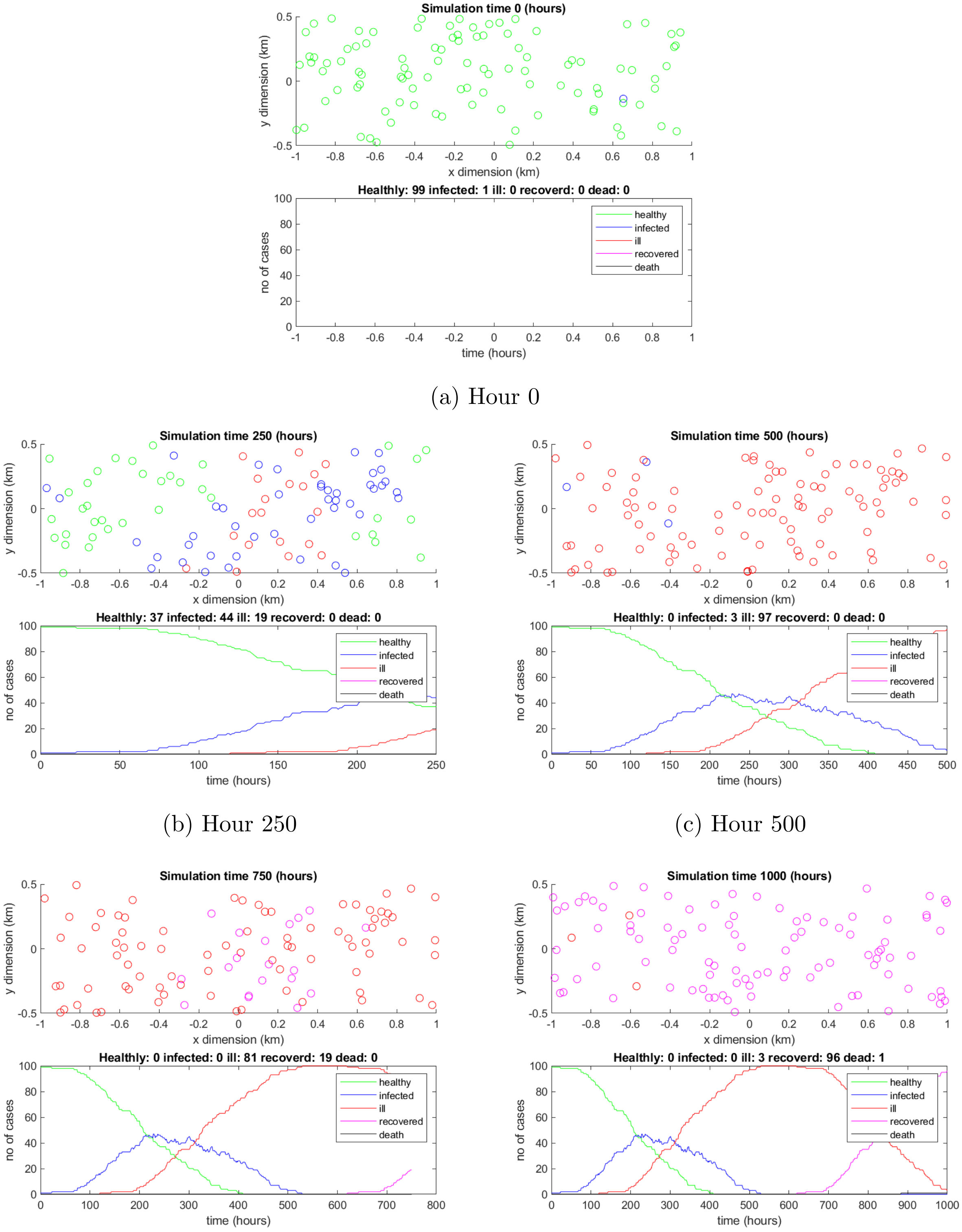
```
%Pandemia simulation using Monte Calro method
%(c) M.Zimnoch, 2020
clear;
clc;
clf;
%Model parameters
Npop = 100;      % Number of population
Xdim = 2;        % Area X dimension in km
Ydim = 1;        % Area Y dimension in km
dx   = 0.05;       % Mean person velocity
MR   = 0.2;       % Ratio of mobile persons
DR   = 0.03;      % Death ratio
dt   = 1;         % Timestep in hours
Tsim = 1000;      % Simulation time
Tinc = 120;       % Incubation mean time
Trec = 500;       % Recovery time
Tinf = 3;         % Contact time required for infection
Rinf = 0.05;     % Contact distance required for infection
%Variables concerning population
Px = rand(Npop,1)*Xdim-Xdim/2; % x coordinate of persons
Py = rand(Npop,1)*Ydim-Ydim/2; % y coordinate of persons
Ps = zeros(Npop,1);      % Status 0-healthy, 1-infected, 2-ill, 3-convalescent, 4 - dead
Pss= zeros(Npop,1);      % Temporary status table
Pa = zeros(Npop,1);      % Status age
Ps(1) = 1;
stat=[];
%Time loop
nt=0;
for t=0:dt:Tsim
    nt=nt+1;
    %calculation of statistics
    stat=[stat;t hist(Ps,[0:4])];
    %visualisation of results
    clf;
    subplot(2,1,1);
    hold on;
    for n=1:Npop
      switch Ps(n)
```

```
         case 0
          plot(Px(n),Py(n),'og');
         case 1
          plot(Px(n),Py(n),'ob');
         case 2
          plot(Px(n),Py(n),'or');
         case 3
          plot(Px(n),Py(n),'om');
      end;
    end;
    axis([-Xdim/2 Xdim/2 -Ydim/2 Ydim/2]);
    title(['Simulation time ' int2str(t) ' (hours)']);
    xlabel('x dimension (km)');
    ylabel('y dimension (km)');
    subplot(2,1,2);
    plot(stat(:,1),stat(:,2),'-g',...
         stat(:,1),stat(:,3),'-b',...
         stat(:,1),stat(:,4),'-r',...
         stat(:,1),stat(:,5),'-m',...
         stat(:,1),stat(:,6),'-k');
     legend('healthy','infected','ill','recovered','death');
     title(['Healthly: ' int2str(stat(nt,2)) ' infected: ' int2str(stat(nt,3)) ' ill: ' int2str(stat(nt,4)) ' recoverd: ' int2str(stat(nt,5)) ' dead: ' int2str(stat(nt,6))]);
     xlabel('time (hours)');
     ylabel('no of cases');
    pause(0.01);
    %moving people
    for n=1:Npop
      if rand(1)< MR && Ps(n) ~= 2
        Px(n) = Px(n)+randn(1)*dx;
        Py(n) = Py(n)+randn(1)*dx;
      end;
      Ix = fix(Px/(0.5*Xdim));
      Iy = fix(Py/(0.5*Ydim));
      Px = Px - Xdim*Ix;
      Py = Py - Ydim*Iy;
    end;
    %counting of infections
    % infection detection
    Pss= zeros(Npop,1);
    for n=1:Npop-1
        for m=n+1:Npop
          %calculation of distance
          distance=sqrt((Px(n)-Px(m))^2+(Py(n)-Py(m))^2);
          if distance<Rinf
            if Ps(m)>0 && Ps(m)<3 && Ps(n)==0
              Pss(n)=1;
            end;
            if Ps(n)>0 && Ps(n)<3 && Ps(m)==0
              Pss(m)=1;
            end;
          end;
        end;
    end;
    % status modification
    for n=1:Npop
        if Pss(n) == 1                   % Increment age of contact with infected person
            Pa(n)=Pa(n)+dt;
        end;
        if Ps(n)>0                       % Increment age of status
            Pa(n)=Pa(n)+dt;
        end;
        if Ps(n)==0 && Pa(n)==Tinf       % If contact > inection time -> infected
            Ps(n)=1;
            Pa(n)=0;
        end;
        if Ps(n)==1 && Pa(n)==Tinc       % If infection > incubation -> ill
```

```
            Ps(n)=2;
            Pa(n)=0;
        end;
        if Ps(n)==2 && Pa(n)==Trec     % If illnes > recovery -> reconvalescent
            if rand(1)>DR
                Ps(n)=3;
            else
                Ps(n)=4;
            end;
            Pa(n)=0;
        end;
    end;
end;
```

It computed and displayed simulation of pandemic for 100 moving people in a space one kilometer by two kilometers. At start only one person was infected, which caused the further spread of infection. People could get infected after being close enough to infected person for set amount of time. Infected people after incubation time have become ill and then they could recover after recovery time or become dead with set death ratio.

For default parameters and 1000 hours (steps) simulation was computed and presented on Figure 1. It shows pandemic without any restrictions of movement, which were introduced in task nr 2. Virus spreaded abruptly causing exponential grow of infected and ill cases. In next step - task nr 1 - different parameters influence on model was tested.



(a) Hour 0

(b) Hour 250

(c) Hour 500

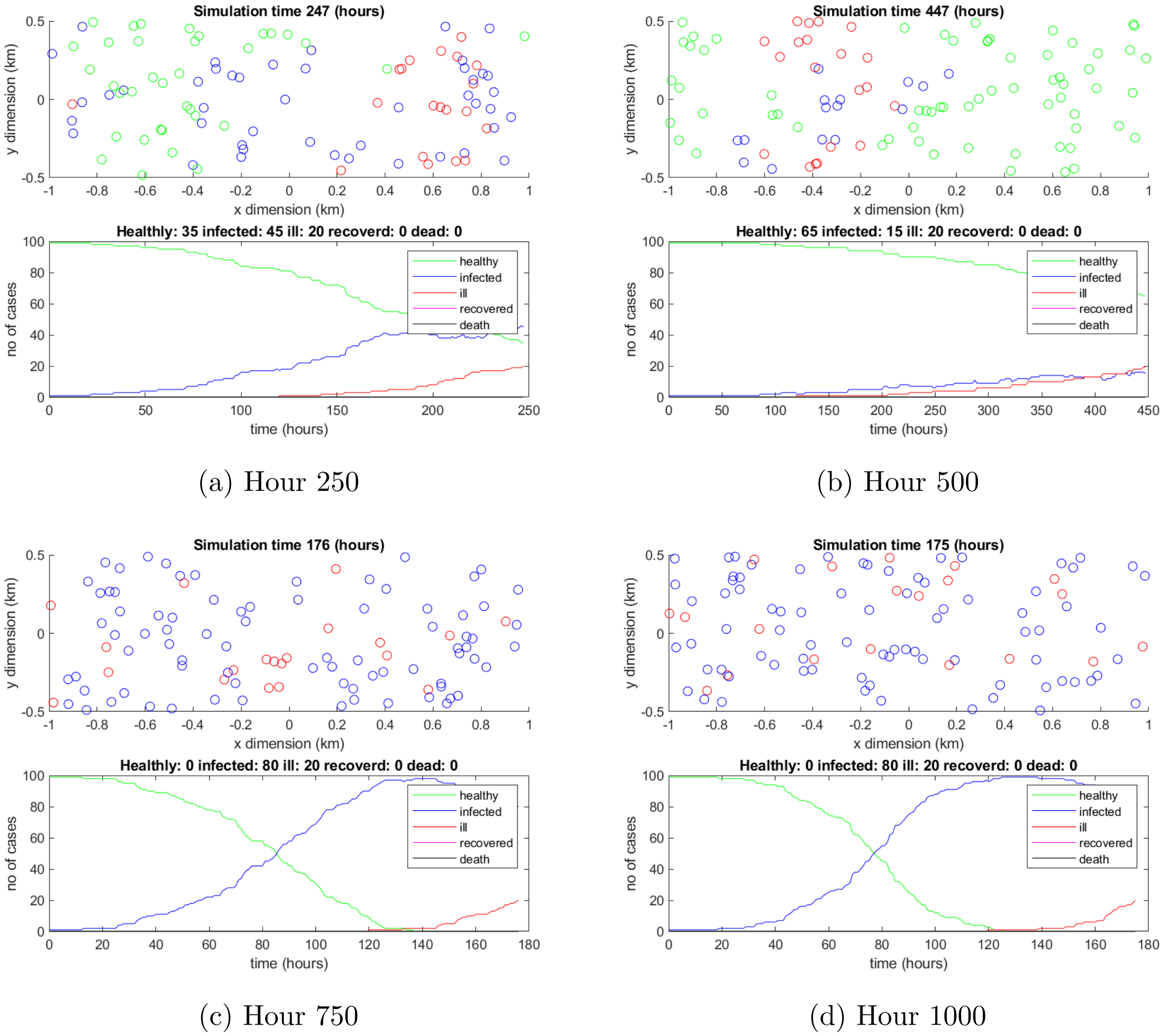
(d) Hour 750

(e) Hour 1000

Figure 1: Simulation for default parameters

2 Checking impact of the model parameters

To test and evaluate influence of every change of parameter in the model, the following condition was introduced: the simulation ended when the ill cases exceeded 20 percent of the total population. The greater the number of steps until this condition is broken, the better, because the growth of the virus is more controlled and less rapid, and therefore less dangerous. For each subfigure in figure 2 and figure 3 title contains changed parameter.



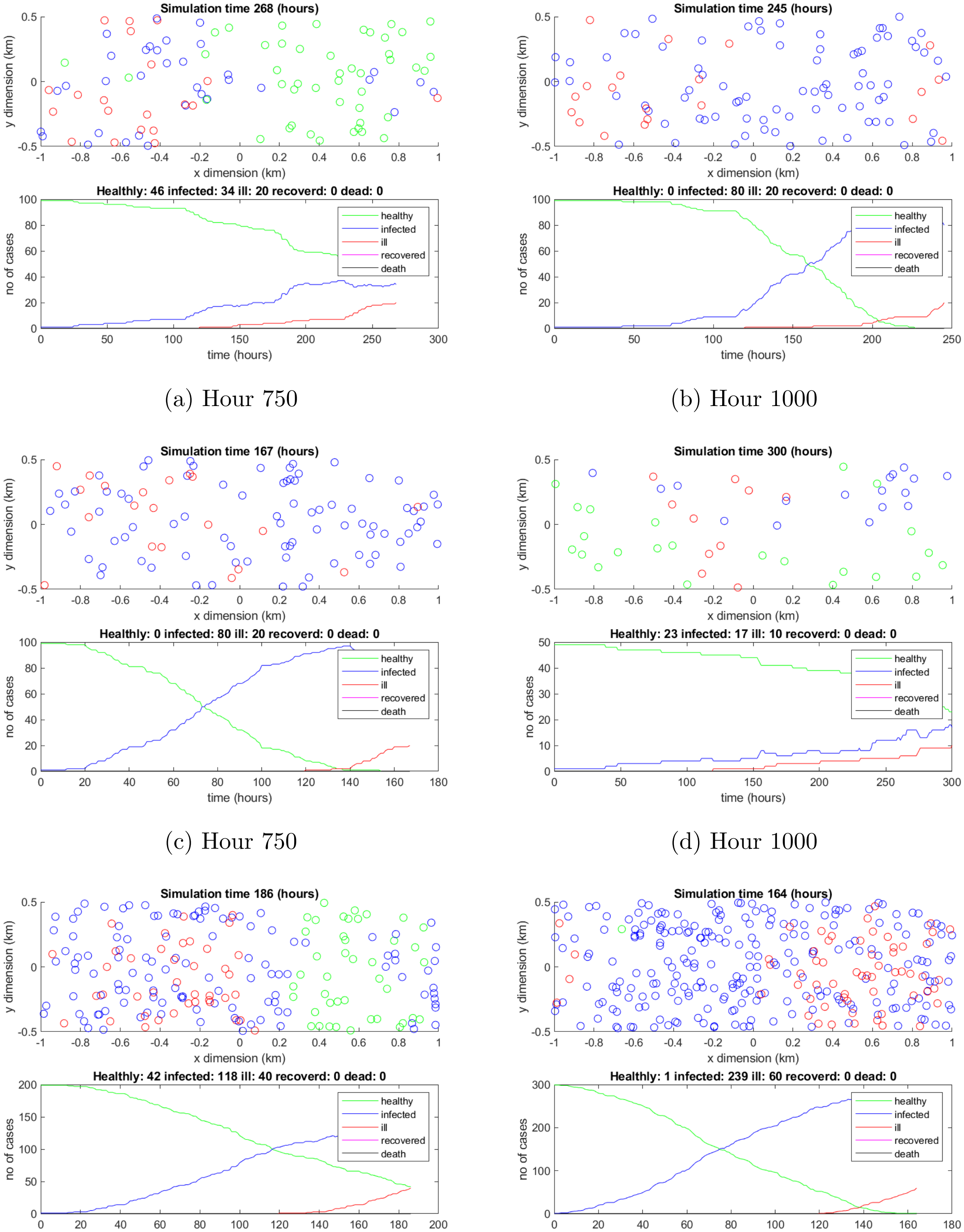
(a) Hour 250

(b) Hour 500

(c) Hour 750

(d) Hour 1000

Figure 2: Simulation for different parameters nr 1



(a) Hour 750

(b) Hour 1000

(c) Hour 750

(d) Hour 1000

(e) Hour 750

(f) Hour 1000

Figure 3: Simulation for different parameters nr 2

Changes in parameters capture differences due to community behavior, such as changes in the number of people who move, the distances they move, or the population density of an area. The change in the number of steps needed to exceed 20 percent of ill cases in the population is shown in the table 1.

Changed parameter	unchanged	dx=0.02	dx=0.1	dx=0.2	MR=0.1	MR=0.5	MR=0.8	Npop=50	Npop=200	Npop=300
Number of steps before breaking ill cases condition	248	448	177	176	269	246	168	301	187	165

Table 1: Dependence of the parameter change on the number of simulation steps

As the data in the table show, changing the model parameters significantly changes the rate of pandemic spread. When increasing parameters that deepen human interaction, such as increasing the people who move (MR), increasing the distance they travel (dx), or increasing the density of people in an area (Npop), the rate of disease increases rapidly. Reducing these parameters, simulating a reduction in social interaction, has a positive effect on limiting pandemic development.

3 Introducing isolation mechanism

For second task isolation mechanism was introduced to the model. Firstly, initial parameters were slightly changed, so that the spread of pandemic would be more visible during whole simulation. Incubation time was reduced to 336 hours which is equal to 2 weeks - time of quarantine. Also population of people was increased to 500, so that pandemic could be more vivid and seen in bigger scale. However density of population didn't change, as dimensions of model were changed accordingly to increased population (Xdim = 4km, Ydim = 2.5km). Secondly and most importantly mechanism of isolation was added, which included testing of people and restricted movement of people in terms of either ill of tested infected. Parameters for testing mechanism consisted of TR - test success rate, Pm - matrix of encounters in last 5 days and Pma - matrix of encounters in last 5 days age.

Restricted movement of people was as follows:

- Ill person does not move at all.
- Infected person that was tested does not move as well.
- Healthy person avoids ill and infected tested people. If next move implies position in area of possible infection, then he/she doesn't move in that hour.
- Infected person that is not tested behaves as if was healthy person.

Tests and encounters mechanism:

- If two people get in distance smaller than contact distance required for infection it is noted in encounter matrix.
- Every step, if encounter took place, age of encounter is updated in encounter age matrix.

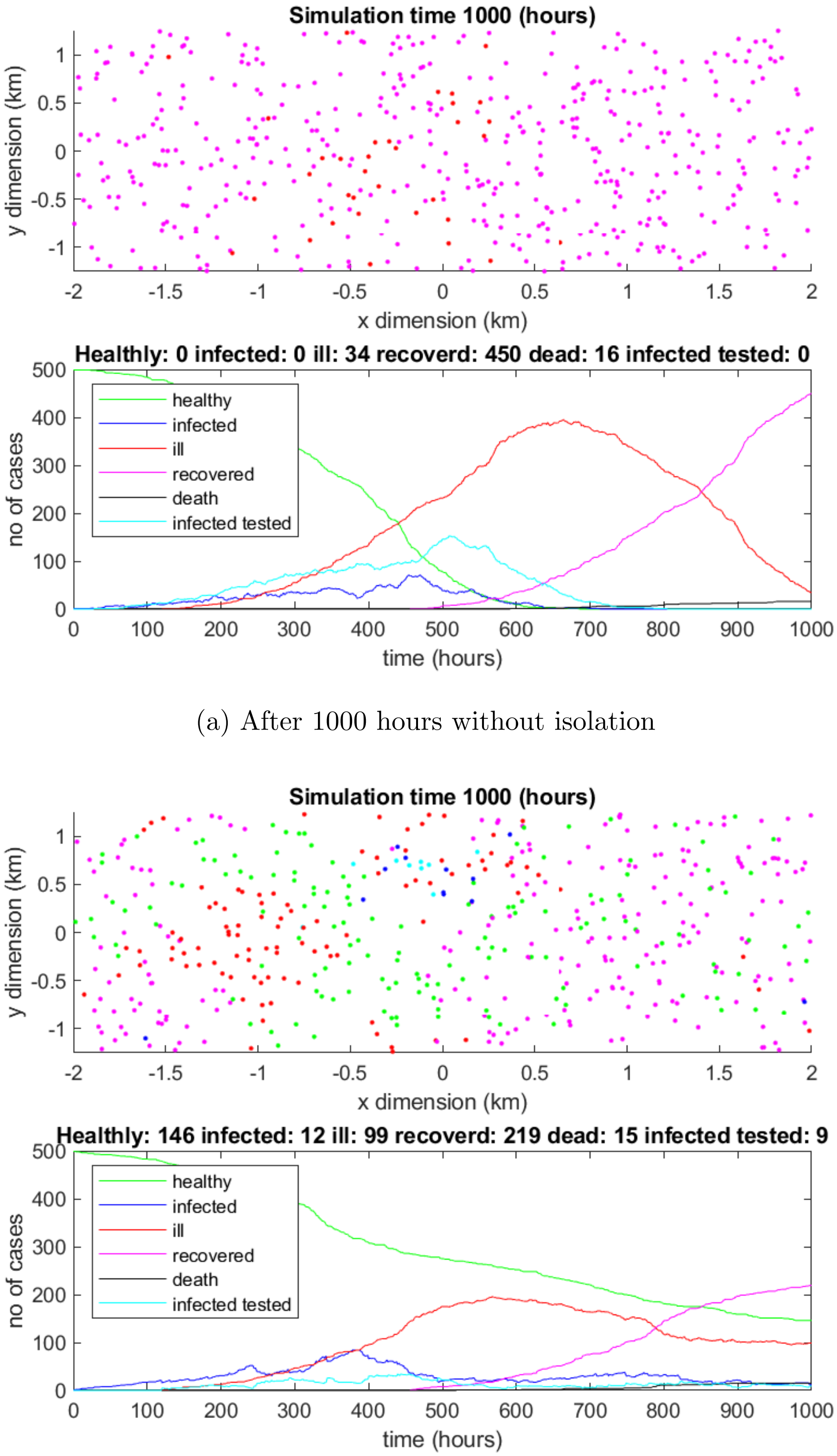
- If 5 days pass encounter and encounter age is erased from matrices.
- When anybody gets ill, everybody that had an encounter with now ill person is tested for virus.
- Test works with set accuracy (in presented simulation it was 95%) and works only for infected people (tests don't work if person encountered infected person, but haven't been infected yet).
- After testing all encounter data for ill person and for tested person is erased.

Code with mentioned mechanisms looks as follows:

```
clc;
clf;

%Model parameters
Npop = 500;      % Number of population
Xdim = 4;        % Area X dimension in km
Ydim = 2.5;        % Area Y dimension in km
dx   = 0.05;       % Mean person velocity
MR   = 0.2;       % Ratio of mobile persons
DR   = 0.03;      % Death ratio
dt   = 1;         % Timestep in hours
Tsim = 1000;      % Simulation time
Tinc = 120;       % Incubation mean time (5 days)
Trec = 336;       % Recovery time (2 weeks)
Tinf = 3;         % Contact time required for infection
Rinf = 0.05;     % Contact distance required for infection
TR = 0.95;        % Test success rate

%Variables concerning population
Px = rand(Npop,1)*Xdim-Xdim/2; % x coordinate of persons
Py = rand(Npop,1)*Ydim-Ydim/2; % y coordinate of persons
Ps = zeros(Npop,1);     % Status 0-healthy, 1-infected, 2-ill, 3-convalescent, 4-dead, 5-infected_tested
Pss = zeros(Npop,1);     % Temporary status table
Pa = zeros(Npop,1);     % Status age
Pm = zeros (Npop, Npop); % People conntacted in last 5 days
Pma = zeros (Npop, Npop); % People conntacted in last 5 days status age
Ps(1) = 1;
stat=[];

%Time loop
nt=0;

for t=0:dt:Tsim
    nt = nt+1;

    %calculation of statistics
    stat = [stat; t hist(Ps, [0:5])];

    %visualisation of results
    clf;
    subplot(2,1,1);
    hold on;

    for n=1:Npop
      switch Ps(n)
        case 0
          plot(Px(n),Py(n),'.g');
        case 1
          plot(Px(n),Py(n),'.b');
        case 2
        plot(Px(n),Py(n),'.r');
       case 3
        plot(Px(n),Py(n),'.m');
       case 5
        plot(Px(n),Py(n),'.c');
      end
    end

    axis([-Xdim/2 Xdim/2 -Ydim/2 Ydim/2]);
    title(['Simulation time ' int2str(t) ' (hours)']);
    xlabel('x dimension (km)');
    ylabel('y dimension (km)');
    subplot(2,1,2);
    plot(stat(:,1),stat(:,2),'-g',...
         stat(:,1),stat(:,3),'-b',...
         stat(:,1),stat(:,4),'-r',...
         stat(:,1),stat(:,5),'-m',...
         stat(:,1),stat(:,6),'-k',...
         stat(:,1),stat(:,7),'-c');
     legend('healthy','infected','ill','recovered','death','infected tested', 'Location','northwest');
     title(['Healthly: ' int2str(stat(nt,2)) ' infected: ' int2str(stat(nt,3)) ' ill: ' int2str(stat(nt,4)) ' recoverd: ' int2str(stat(nt,5)) ' dead: ' int2str(stat(nt,6)) ' infected tested: ' int2str(stat(nt, 7))]);
     xlabel('time (hours)');
     ylabel('no of cases');
    pause(0.01);

    % moving people
    for n=1:Npop
      if Ps(n) ~= 2 && Ps(n) ~= 4 && Ps(n) ~= 5 && rand(1) < MR  % condition states that person does not moves if he/she knows that is infected/ill (so ill cases and infected tested) and with MR probability
        pdx = randn(1);
        pdy = randn(1);
        Px_temp = Px(n)+pdx*dx;
        Py_temp = Py(n)+pdy*dx;
        safe_move = true;

        if Ps(n) == 0 || Ps(n) == 1 % this condition prevents healthy person from stepping in area where he/she can get infected
            for k=1:Npop
                if Ps(k) == 2 || Ps(k) == 5 % checks distance to ill/infected person
                    distance = sqrt((Px_temp-Px(k))^2+(Py_temp-Py(k))^2);
                    if distance <= Rinf  % if distance is to small then it is not safe to move
                        safe_move = false;
                        break;
                    end
                end
            end
        end

        % if move is, in respect to whole population, not dangerous then current person can move that way
        if safe_move == true
            Px(n) = Px_temp;
            Py(n) = Py_temp;
        end
      end

      Ix = fix(Px/(0.5*Xdim));
      Iy = fix(Py/(0.5*Ydim));
      Px = Px - Xdim*Ix;
      Py = Py - Ydim*Iy;
    end

    %counting of infections
    % infection detection
    Pss = zeros(Npop,1);

    for n=1:Npop-1
        for m=n+1:Npop
          %calculation of distance
          distance = sqrt((Px(n)-Px(m))^2+(Py(n)-Py(m))^2);

          if distance < Rinf
            if ((Ps(m)>0 && Ps(m)<3) || Ps(m)==5) && Ps(n)==0  % added infections coming from tested people
              Pss(n)=1;
            end
            if ((Ps(n)>0 && Ps(n)<3) || Ps(n)==5) && Ps(m)==0  % added infections coming from tested people
              Pss(m)=1;
            end

            Pm(n, m) = 1; % if two people get close enough it is saved as encounter that could be used in potential situation of illness to know who should be tested
            Pm(m, n) = 1;
          end
        end
    end

    % status modification
    for n=1:Npop
        if Pss(n) == 1                  % Increment age of contact with infected person
            Pa(n)=Pa(n)+dt;
        end

        if Ps(n)>0                      % Increment age of status
            Pa(n)=Pa(n)+dt;
        end

        if Ps(n)==0 && Pa(n)==Tinf      % If contact > inection time -> infected
            Ps(n)=1;
            Pa(n)=0;
        end

        if Ps(n)==1 && Pa(n)==Tinc      % If infection > incubation -> ill
            Ps(n)=2;
            Pa(n)=0;

            % if somebody gets ill every person that have encountered that person is tested
            for k=1:Npop
                if (Ps(k) == 1 || Ps(k) == 0) && Pm(n, k) == 1
                    if rand(1) < TR && Ps(k) == 1  % test has certain rate success
                        Ps(k) = 5;
                    end
                    Pm(n, k) = 0; % after test encounter data is erased
                    Pma(n, k) = 0;
                end
            end

            for p=1:Npop
                Pm(n, p) = 0; % after test encounter data is erased for every encounter with person who got ill
                Pma(n, p) = 0;
            end
        end

        if Ps(n)==5 && Pa(n)==Tinc      % If infection tested > incubation -> ill
            Ps(n)=2;
            Pa(n)=0;
        end

        if Ps(n)==2 && Pa(n)==Trec      % If illnes > recovery -> reconvalescent
```

```
            if rand(1)>DR
                Ps(n)=3;
            else
                Ps(n)=4;
            end
            Pa(n)=0;
        end
    end

    for n=1:Npop % for each encounter in encounter matrix its age is increased and if it exceed 5 days then its erased
        for m=1:Npop
          if Pm(n, m) == 1
            Pma(n, m) = Pma(n, m) + dt;
          end

          if Pm(m, n) == 1
            Pma(m, n) = Pma(m, n) + dt;
          end

          if Pma(n, m) > 120
            Pma(n, m) = 0;
            Pm(n, m) = 0;
          end

          if Pma(m, n) > 120
            Pma(m, n) = 0;
            Pm(m, n) = 0;
          end
        end
    end
end
```

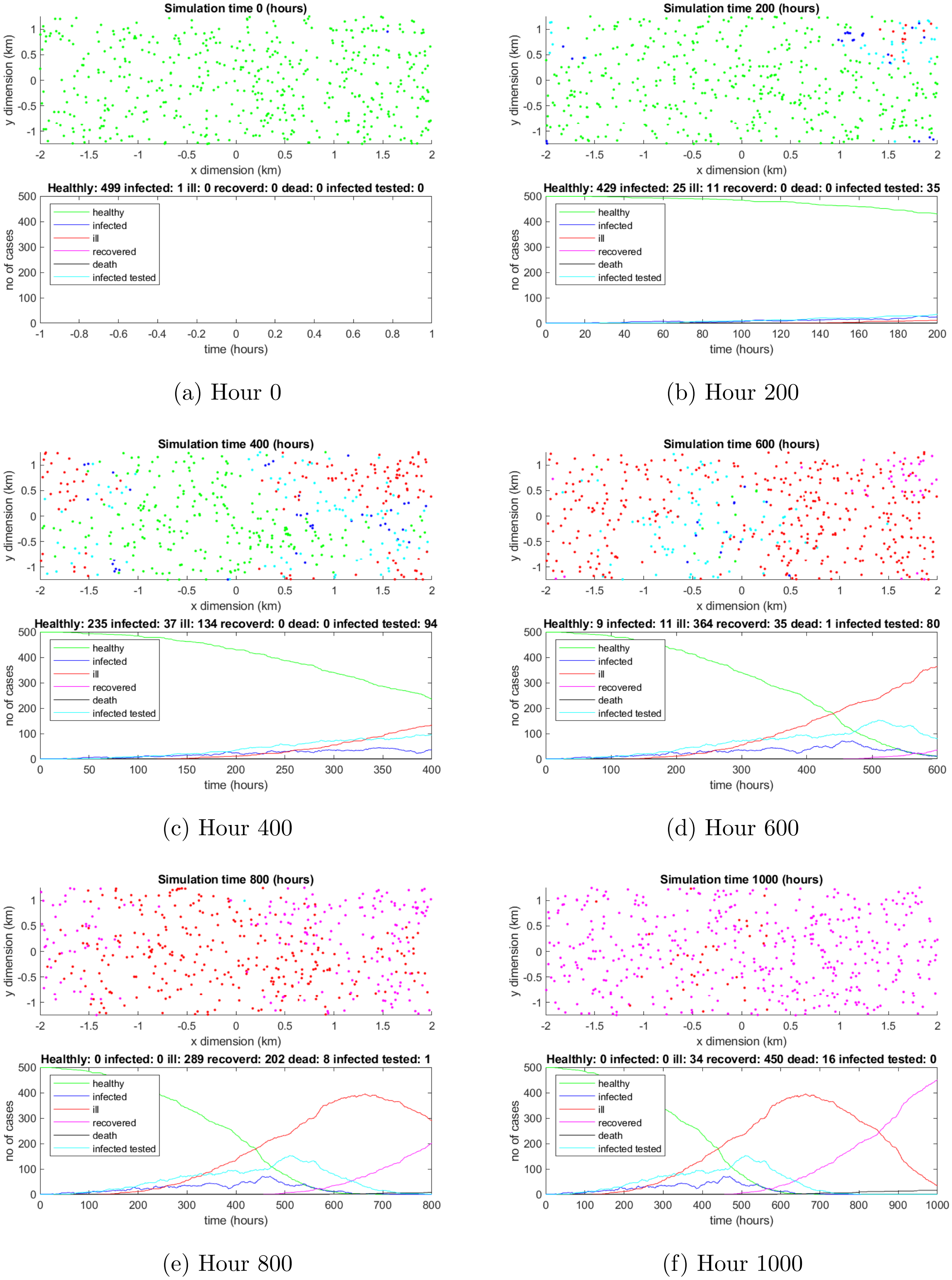
Simulation was made using bigger population and dimensions for model both with and without isolation and test mechanisms. The main goal was to flatten the curve of ill cases, which would relieve the burden on hospitals and the health care system. Simulation for model without mechanisms is showed on figure 5. For model with mechanisms, showed on figure 6, number of steps was extended to 1400 due to slow propagation of pandemic. mechanism. After 1000 steps (1000h hours) figures showed 4 were obtained, which shows impact of introduced mechanisms and difference between models..



(a) After 1000 hours without isolation

(b) After 1000 hours with isolation

Figure 4: Influence of isolation mechanism on model



(a) Hour 0

(b) Hour 200

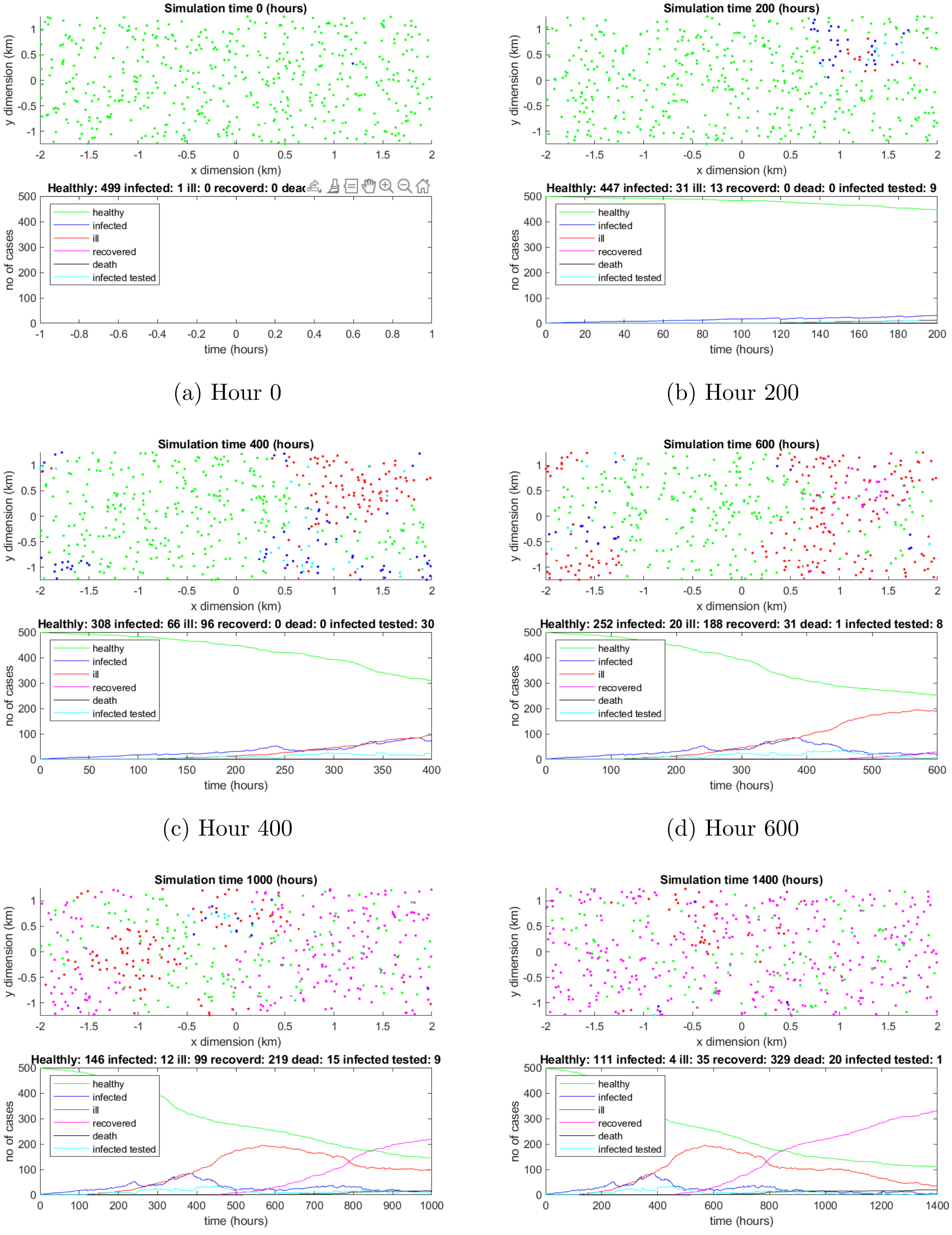
(c) Hour 400

(d) Hour 600

(e) Hour 800

(f) Hour 1000

Figure 5: Simulation for model without added mechanisms



(a) Hour 0

(b) Hour 200

(c) Hour 400

(d) Hour 600

(e) Hour 1000

(f) Hour 1400

Figure 6: Simulation for model with added mechanisms

Peek of ill cases for model without isolation and testing mechanisms exceeds 400 people (which is over 80% of population), while the same peek for model with added mechanisms is below 200 people (less than 40% of population). Moreover, after 1400 hours in simulation with added mechanisms, there was significant number of healthy people, never infected, in opposite to simulation without these mechanisms.

4 Conclusions

During task number 1, when parameters such as percentage of moving people, length of single person step or density of people were changed, it turned out that reducing opportunities for person-to-person contact had a positive effect on pandemic suppression. For this reason, in task number 2, a mechanism of testing and isolation was introduced concerning ill and tested infected cases. The results of this simulation were very promising due to the more than doubling of the maximum number of people who were sick at any one time and would potentially take up space in hospitals. Thus, the hypothesis that introducing isolation of sick people and testing potentially sick people has a positive effect on preventing the mass spread of a pandemic was confirmed.