

Simulation of the Covid-19 pandemic data series

George Daniel Mateescu¹

Abstract. *In the context of the Covid-19 pandemic, we set out to develop a model to simulate the spread of the pandemic, a model based on current information on the process of spreading SARS-CoV-2 virus infection. The algorithm is made on days, every day there is a number of infected (sick) a number of contacts and a probability of contagion / transmission, which results in the coefficient R of disease spread.*

Keywords: pandemic simulation, probability of contagion

JEL Classification: C15

Algorithm / model

The simulation algorithm aims to reproduce the evolution of the daily pandemic. Thus, a population that is exposed to the virus is considered. Every day a number of individuals become infected. Each infected individual can pass the disease on, and after a while it either heals or disappears from the system. From the point of view of evolution, healing or disappearance are equivalent in the sense that such an individual can no longer transmit the disease.

The current day is d , with the initial value $d = 1$

On day d :

The entire population is traversed by successive individuals

If the current individual is infected between the 5th and 14th day after infection, then

Randomly choose "c" individuals

¹ Institute for Economic Forecasting, Romanian Academy, e-mail dan@ipe.ro

For each individual it generates a random number in (0,1)

If the number generated is less than the threshold then:

If it is uninfected, then it will be infected

Go to the next day, $d = d + 1$, and resume the d-day algorithm

The combination of factors c and threshold allows the simulation of any factor R by the formula $R = c * \text{threshold}$. For example, $c = 3$ and $\text{threshold} = 0.3$ determine a coefficient $R = 0.9$

Implementation of the algorithm in C ++ language

```
// Includes the required libraries
```

```
#include <math.h>
```

```
#include <stdio.h>
```

```
#include <stdint.h>
```

```
#include <time.h>
```

```
// The program
```

```
int main ()
```

```
{
```

```
    FILE * f;
```

```
// the file that contains, per days, the number of patients (active infected)
```

```
    int pop[200000][2];
```

```
    int popd[200000][2];
```

The indexed variables `pop` and `popd` represent a population coding, in two consecutive days. Each individual is represented by `pop[i][]` (respectively `popd[i][]`) having two `pop [i] [0]` attributes which has the value 0 if the individual is uncontaminated, the value 1 if it is contaminated and the value 2 if a passed through infection, ie more than 14 days have passed since the contamination. The `pop [i] [1]` component contains the day of contamination; the model starts from a conventional value 1, for day 1 and continues up to a default value (default number of days). The combination of the two attributes allows the representation of active cases, for example between the 5th and the 14th day after contamination. It also covers the case of individuals who have gone through the infection, respectively have passed the 14th day after contamination. The model does not include deaths, these being included in the crowd of those who went through infection. In this way, the model is oriented only towards the spread of the infection.

```

long int i; // population counter

long int j = 1; //counter

long int d = 1; // d - days, starting from day 1

long int n = 1; // n - the number of active infected on day z, initially 1

long int k; // k - a randomly selected individual, contact likely to be contaminated

long int c; // forward to c contacts

float cf; // probability of transmission, of contagion

float threshold;

```

The combination of parameters `c` - number of contacts and `threshold` - the probability of contagion (the probability with which an infected person transmits the virus to another contact) can simulate the transmission coefficient `R`, respectively how many infected people result from an infected person. For example, 3 contacts and the threshold of 0.4 represent a coefficient $R = 1.2$

```

f = fopen ("/ root / date", "w +");

// associate the file that will contain the number of infected in each day

```

```

    srand (time (NULL));
// initialize the random number generator
    long int p;
    printf ("% s", "number of people:");
    scanf ("% i", & p);
// the number of persons (population) is a parameter, having the value contained in the
variable p
    for (i = 0; i <p; i ++) {pop[i][0] = 0; pop[i][1] = 0; popd[i][0] = 0; popd[i][1] = 0;};
// resets the population, ie completely uninfected
    pop[0][0] = 1; pop[0][1] = 1; popd[0][0] = 1; popd[0][1] = 1;
// character 0, contaminated on day 1
    printf ("% s", "number of direct contacts =");
    scanf ("% i", & c); // parameter, the number of direct contacts of an infected person
// initializes the number of direct contacts of an infected person
    printf ("% s", "probability of an infected person having direct contacts =");
    scanf ("% f", & threshold); // the probability of transmitting the infection
// initializes the threshold (probability) by which a contagion transmits the virus to a
contact
    while (d <1000) // simulates, for example, 1000 days!
    {
        for (j = 0; j <p; j ++)
            {// browse the population
                if (pop[j][0] == 1 && d > (pop[j][1] +5) && d <(pop[j][1] +14))
                    {

```

```

// if an infected person is between day 5 and day 14 after contamination
    for (i = 0; i <c; i ++ )
    {
        cf = ((float) (rand ()% p)) / ((float) p);
// generates a random number between 0 and 1
        if (cf <threshold) // with the chosen probability
    {
        k = rand ()% p;
// choose k, ie a randomly chosen contact from the population
        if (popd[k][0] == 0)
        {
// if it is uninfected, then it will be infected on day z
            popd[k][0] = 1;
            popd[k][1] = z;

            };

            };

            };

        };

        fprintf (f, "% 4i \n", n);
// deposit in the file the number of infected, active people

```

```
// at the end of the day count how many active infected people there are; will be
memorized the next day
```

```
n = 0;
```

```
    for (j = 0; j <p; j ++)
```

```
        {
```

```
            if (popd [j] [0] == 1)
```

```
                {
```

```
                    n ++;
```

```
                    if (popd[j][1] <(d-14)) popd[j][0] = 2;
```

```
// if 14 days have passed, I get out of the active situation
```

```
                };
```

```
        };
```

```
    d ++;
```

```
    for (j = 0; j <p; j ++) {pop[j][0] = popd[j][0]; pop[j][1] = popd[j][1];};
```

```
};
```

```
fclose (f);
```

```
return 1;
```

```
}
```

After executing the program, the output data file is transferred to an Excel worksheet, which allows the analysis of the entire data set, as well as the evolution of the pandemic by day.

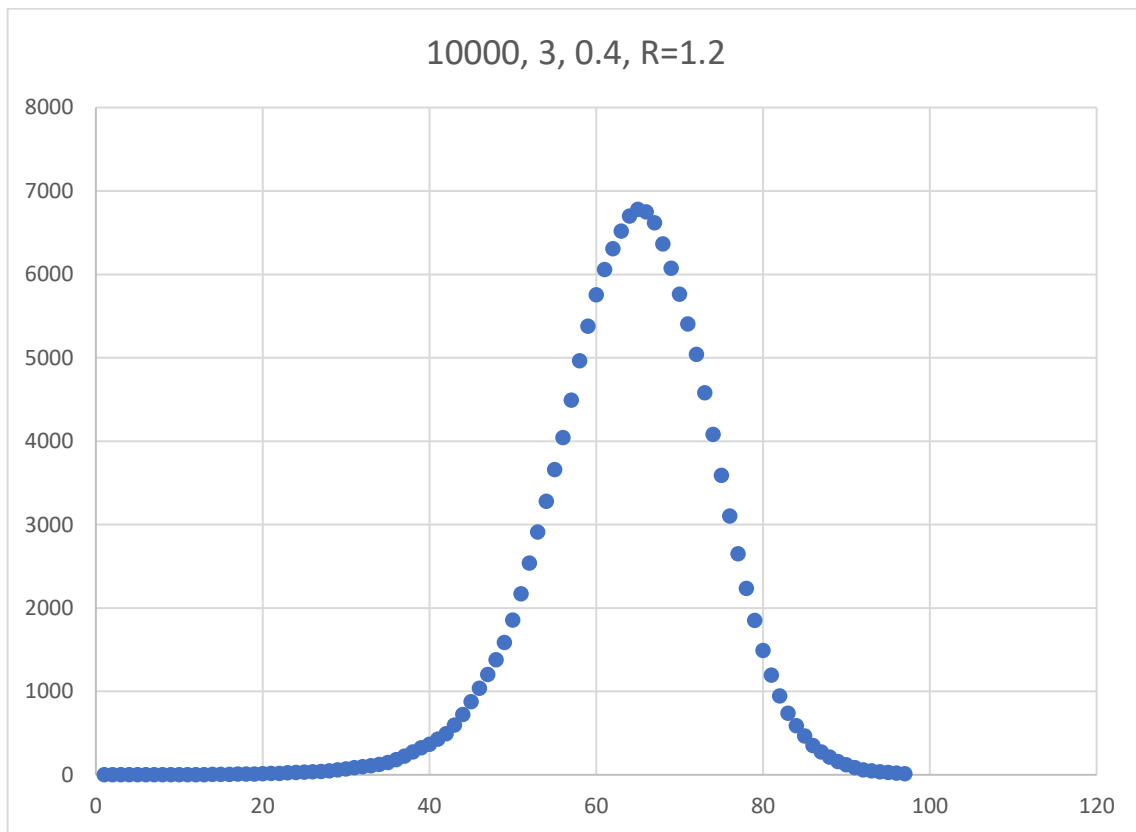
Simulation results

Population 10000

Contacts 3

Threshold 0.4

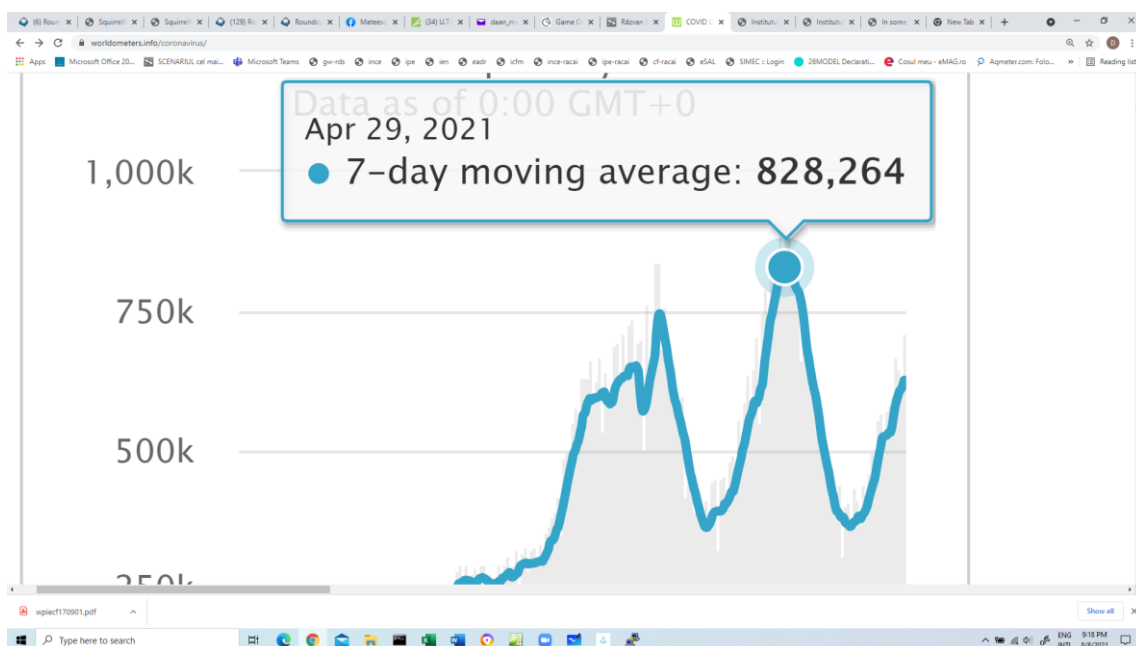
R 1.2



It is observed that in about 80 days the whole population is cold by infection, with a maximum of almost 8000 infected on day 55.

The sum of all the infected, divided by the active period of 14 days, gives approximately the entire population, which justifies the statement that practically the entire population goes through infection.

The comparison with the real situation of the world epidemic makes us consider that the simulations are correct. Basically, we have a complete similarity between the simulations and the statistical data recorded in the third wave in January-May 2020.



Source: <https://www.worldometers.info/coronavirus/>

References (sources)

1. <https://www.worldometers.info/coronavirus/>
2. <https://www.who.int/> (World Health Organization)
3. <https://www.ema.europa.eu/en> (European Medicines Agency)
4. <https://www.cdc.gov/> (Centers for Disease Control and Prevention)