A Mathematical model for the diffusion of Coronavirus 19 among the Italian population

Paolo Maccallini*

March 18, 2020

maccallini.paolo@gmail.com

Abstract

In this paper, I propose a distribution for the number of infected subjects in Italy during the outbreak of Coronavirus 19. To do that I find a logistic curve for the number of deaths due to this virus, in Italy. I also use a density of probability for the fatality rate and one for the number of days from infection to death. These functions have been built using recently published statistical data on the Chinese outbreak.

1 Introduction

A series of cases of pneumonia reported in the city of Wuhan (Cina) since December 2019, has led to the discovery of a new human virus, initially named *2019 novel coronavirus* (2019-nCoV) by the World Health Organization (WHO). 2019-nCoV causes acute respiratory syndrome, called *coronavirus disease 2019* (COVID-19). On 11 February 2020, the name of this new virus was changed into SARS-CoV-2 (1). It belongs to the subgenus *botulinum* within the family *Coronaviridae* (2). It is a zoonotic disease that can be transmitted within our species. It has been suggested that bats and snakes are possible reservoirs of SARS-CoV-2 (3).

2 Mortality rate

A recent review of the available literature has reported a mortality rate with a mean value of 0.07 and a 95% CI equal to $(0.04 - 0.10)$ (4). To build a discrete distribution for the random variable M I have considered a binomial distribution $B(n_M, p_M)$:

Eq. 1 $$P(M = m_k) = p_M^k (1 - p_M)^{n_M - k} \binom{n_M}{k}, \ \forall k = 0, 1, 2, \dots, n_M$$

where m_k is the number of individuals who succumb to the disease in a group of 100 individuals infected by the virus. If σ_M is its standard deviation, we have

$$\begin{cases} \mu_M = n_M p_M \\ \sigma_M^2 = n_M p_M (1 - p_M) \end{cases} \Rightarrow \begin{cases} n_M = \dfrac{\mu_M}{p_M} \\ \sigma_M^2 = \dfrac{\mu_M}{p_M} p_M (1 - p_M) \end{cases} \Rightarrow \begin{cases} n_M = \dfrac{\mu_M}{p_M} \\ \sigma_M^2 = \mu_M (1 - p_M) \end{cases} \Rightarrow \begin{cases} n_M = \dfrac{\mu_M}{p_M} \\ p_M = 1 - \dfrac{\sigma_M^2}{\mu_M} \end{cases} \Rightarrow$$

Eq. 2 $$\begin{cases} n_M = \frac{\mu_M}{1 - \frac{\sigma_M^2}{\mu_M}} \\ p_M = 1 - \frac{\sigma_M^2}{\mu_M} \end{cases}$$

We know that $\mu_M = 7$ and that there is a probability of 0.95 that $m_k \in [4,10]$. The latter means that the following equation has to be true:

Eq. 3 $$P(4 \leq M = m_k \leq 10) = 0.95 \Leftrightarrow \sum_4^{10} p_M^k (1 - p_M)^{n_M - k} \binom{n_M}{k} = 0.95$$

If we consider that $p_M = \mu_M / n_M$, then we have

Eq. 4 $$\sum_4^{10} \left(\frac{\mu_M}{n_M}\right)^k \left(1 - \frac{\mu_M}{n_M}\right)^{n_M - k} \binom{n_M}{k} = 0.95$$

This equation can be solved numerically by imposing increasing integer values to n_M and then searching for the following minimum:

Eq. 5 $$\min_{n_M = 11, 12, \dots} \left\{ \left| \sum_4^{10} \left(\frac{\mu_M}{n_M}\right)^k \left(1 - \frac{\mu_M}{n_M}\right)^{n_M - k} \binom{n_M}{k} - 0.95 \right| \right\}$$

This calculation gives $n_M = 13$ and $p_M = 0.53846$. The distribution for M is reported in Figure 1.

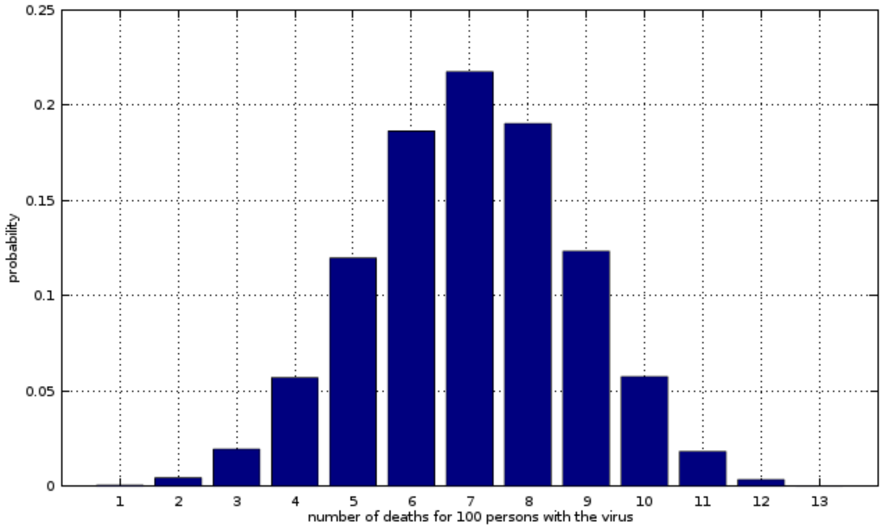

Figure 1. Binomial distribution of the random variable M.

Another possible way to find the distribution for M would be to search for a Gamma law $\Gamma(\alpha_M, \lambda_M)$:

Eq. 6 $$f_M(m) = \begin{cases} \frac{\lambda_M^{\alpha_M}}{\Gamma(\alpha_M)} m^{(\alpha_M - 1)} e^{-\lambda_M m} & \forall\, m > 0 \\ 0 & \forall\, m \leq 0 \end{cases}$$

where

Eq. 7 $$\begin{cases} \mu_M = \frac{\alpha_M}{\lambda_M} \\ \sigma_M^2 = \frac{\alpha_M}{\lambda_M^2} \end{cases} \Rightarrow \begin{cases} \mu_M = \sigma_M^2 \lambda_M \\ \alpha_M = \sigma_M^2 \lambda_M^2 \end{cases} \Rightarrow \begin{cases} \lambda_M = \frac{\mu_M}{\sigma_M^2} = \frac{7}{\sigma_M^2} \\ \alpha_M = \frac{\mu_M^2}{\sigma_M^2} = \frac{49}{\sigma_M^2} \end{cases}$$

To find σ_M we must ask for the following equation to be verified:

$$\int_4^{10} f_M(m) dm = 0.95 \Leftrightarrow \frac{\left(\frac{7}{\sigma_M^2}\right)^{\alpha_M}}{\Gamma\left(\frac{49}{\sigma_M^2}\right)} \int_4^{10} m^{(\alpha_M - 1)} e^{-\lambda_M m} dm = 0.95$$

This problem can be solved assigning increasing values to σ_M and then searching for the following minimum:

Eq. 8 $$\min_{n_M = 11,12,\ldots} \left\{ \left| \frac{\left(\frac{7}{\sigma_M^2}\right)^{\alpha_M}}{\Gamma\left(\frac{49}{\sigma_M^2}\right)} \int_4^{10} m^{(\alpha_M - 1)} e^{-\lambda_M m} dm - 0.95 \right| \right\}$$

The integral can be calculated numerically (for instance with the Simpson rule). This calculation gives $\sigma_M = 1.7$ and thus:

Eq. 9 $$\begin{cases} \lambda_M = \frac{\mu_M}{\sigma_M^2} = 2.4221 \\ \alpha_M = \frac{\mu_M^2}{\sigma_M^2} = 16.955 \end{cases}$$

The probability density and the repartition function are plotted in Figure 2.

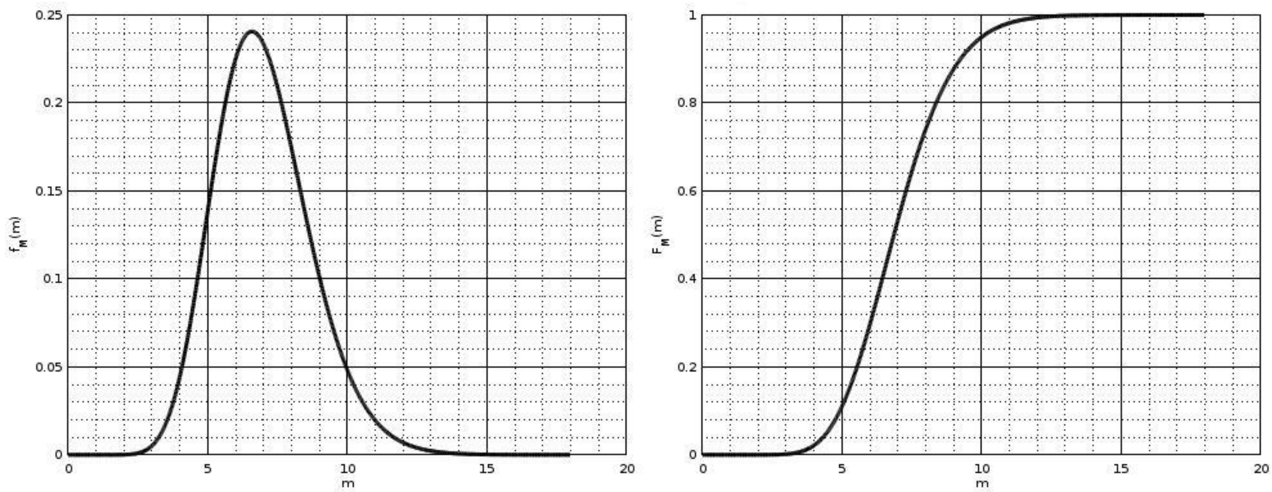

Figure 2. Gamma distribution of the random variable M.

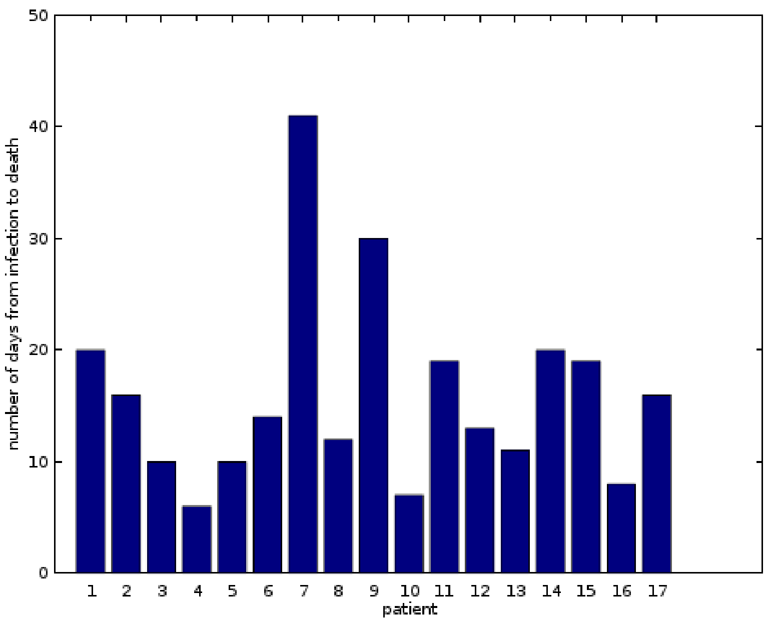

Figure 3. Time from infection to death for 17 patients (10).

3 Time from infection to death

Let's name Δ the random variable that represents the number of days between the first contact with the virus and the death. A study reported the value of this parameter for 17 cases (10) (Figure 3). From this sample, we calculate a mean $\mu_\Delta = 16\ days$ and a standard deviation $\sigma_\Delta = 8.81\ days$. As can be easily seen, these parameters don't fit a binomial distribution nor a Poisson's one. If we assume that Δ has a Gamma law, we have

Eq. 10 $$f_\Delta(\delta) = \begin{cases} \frac{\lambda_\Delta^{\alpha_\Delta}}{\Gamma(\alpha_\Delta)} \delta^{(\alpha_\Delta - 1)} e^{-\lambda_\Delta \delta} & \forall\, \delta > 0 \\ 0 & \forall\, \delta \leq 0 \end{cases}$$

where

Eq. 11 $$\begin{cases}\mu_\Delta = \frac{\alpha_\Delta}{\lambda_\Delta} \\ \sigma_\Delta^2 = \frac{\alpha_\Delta}{\lambda_\Delta^2}\end{cases} \Rightarrow \begin{cases}\mu_\Delta = \sigma_\Delta^2 \lambda_\Delta \\ \alpha_\Delta = \sigma_\Delta^2 \lambda_\Delta^2\end{cases} \Rightarrow \begin{cases}\lambda_\Delta = \frac{\mu_\Delta}{\sigma_\Delta^2} = 0.20614 \\ \alpha_\Delta = \frac{\mu_\Delta^2}{\sigma_\Delta^2} = 3.2983\end{cases}$$

The diagram of $f_\Delta(\delta)$ and of its distribution function are plotted in Figure 4.

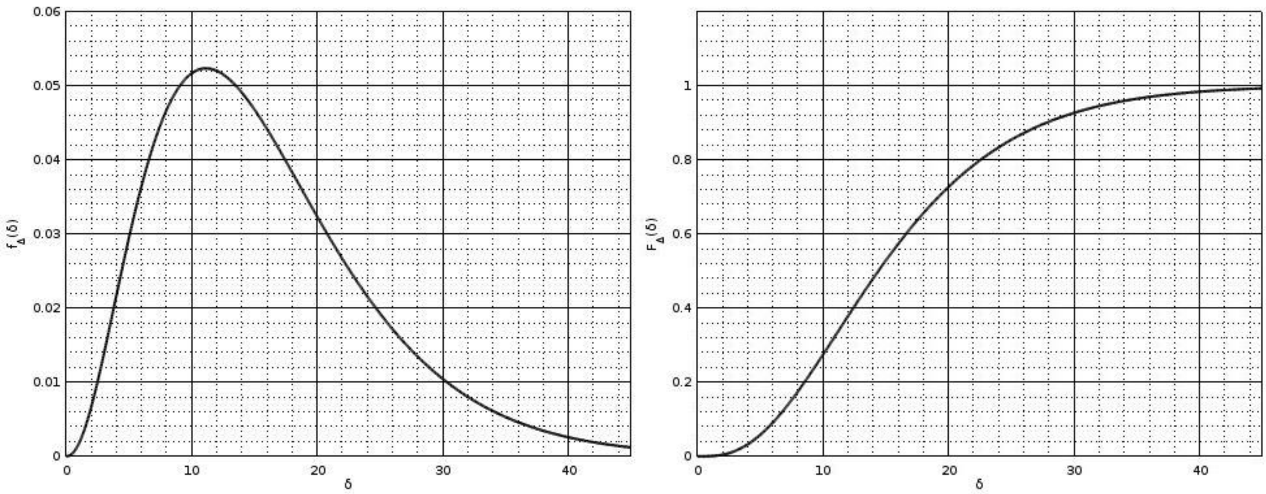

Figure 4. The density of probability (left) and distribution function (right) for Δ.

4 Number of deaths

The total number of deaths is the most reliable measure and we shall use it to make a prevision about the number of infected people. It seems reasonable to assume that the total number of deaths at time *t* is a solution of the following differential equation:

Eq. 12 $$\frac{df(t)}{dt} = f(t)\left(r - \frac{r}{k} f(t)\right)$$

where r, k are defined as follows (6). If we indicate $\lambda(t)$ the rate of infected people and $\mu(t)$ the rate of recovery it seems reasonable to write:

Eq. 13 $$\begin{cases}\lambda(t) = a_1 - b_1 f(t) \\ \mu(t) = a_2 + b_2 f(t)\end{cases}$$

with $a_1, a_2 > 0$ and $b_1, b_2 \geq 0$. Then we have

$$\frac{df(t)}{dt} = [\lambda(t) - \mu(t)]f(t) = [a_1 - a_2 - (b_1 + b_2) f(t)]f(t)$$

If we now define $r = a_1 - a_2$ and $k = \frac{a_1 - a_2}{b_1 + b_2}$, we have Eq. 12, whose solution for $b_1, b_2 \neq 0$ is

Eq. 14 $$f(t) = \frac{k}{1 + \left(\frac{k}{f_0} - 1\right) e^{-r(t - t_0)}}$$

Let's assume now that we know experimentally the following values: (t_0, f_0), (t_1, f_1), (t_2, f_2). How can we calculate k, r? We must solve the following non-linear system:

Eq. 15
$$\begin{cases}\psi_1(r,k) = f_1\left[1+\left(\frac{k}{f_0}-1\right)e^{-r(t_1-t_0)}\right]-k=0\\ \psi_2(r,k) = f_2\left[1+\left(\frac{k}{f_0}-1\right)e^{-r(t_2-t_0)}\right]-k=0\end{cases}$$

By observing that

$$\frac{\partial\psi_1}{\partial r} = -f_1(t_1-t_0)\left(\frac{k}{f_0}-1\right)e^{-r(t_1-t_0)}, \frac{\partial\psi_1}{\partial k} = f_1\frac{e^{-r(t_1-t_0)}}{f_0}-1$$

$$\frac{\partial\psi_2}{\partial r} = -f_2(t_2-t_0)\left(\frac{k}{f_0}-1\right)e^{-r(t_2-t_0)}, \frac{\partial\psi_2}{\partial k} = f_2\frac{e^{-r(t_2-t_0)}}{f_0}-1$$

we have that the Jacobean of the system in Eq. 15 is given by:

$$J(r,k) = \begin{vmatrix}\frac{\partial\psi_1}{\partial r} & \frac{\partial\psi_1}{\partial k}\\ \frac{\partial\psi_2}{\partial r} & \frac{\partial\psi_2}{\partial k}\end{vmatrix} = \begin{vmatrix}-f_1(t_1-t_0)\left(\frac{k}{f_0}-1\right)e^{-r(t_1-t_0)} & f_1\frac{e^{-r(t_1-t_0)}}{f_0}-1\\ -f_2(t_2-t_0)\left(\frac{k}{f_0}-1\right)e^{-r(t_2-t_0)} & f_2\frac{e^{-r(t_2-t_0)}}{f_0}-1\end{vmatrix} =$$

$$= -\frac{f_1(t_1-t_0)\left(\frac{k}{f_0}-1\right)\left[f_2\frac{e^{-r(t_2-t_0)}}{f_0}-1\right]}{e^{r(t_1-t_0)}} + \frac{f_2(t_2-t_0)\left(\frac{k}{f_0}-1\right)\left[f_1\frac{e^{-r(t_1-t_0)}}{f_0}-1\right]}{e^{r(t_2-t_0)}} \Rightarrow$$

Eq. 16
$$J(r,k) = \left(\frac{k}{f_0}-1\right)e^{-r}\left\{\frac{f_2(t_2-t_0)\left[f_1\frac{e^{-r(t_1-t_0)}}{f_0}-1\right]}{e^{(t_2-t_0)}} - \frac{f_1(t_1-t_0)\left[f_2\frac{e^{-r(t_2-t_0)}}{f_0}-1\right]}{e^{-(t_1-t_0)}}\right\}$$

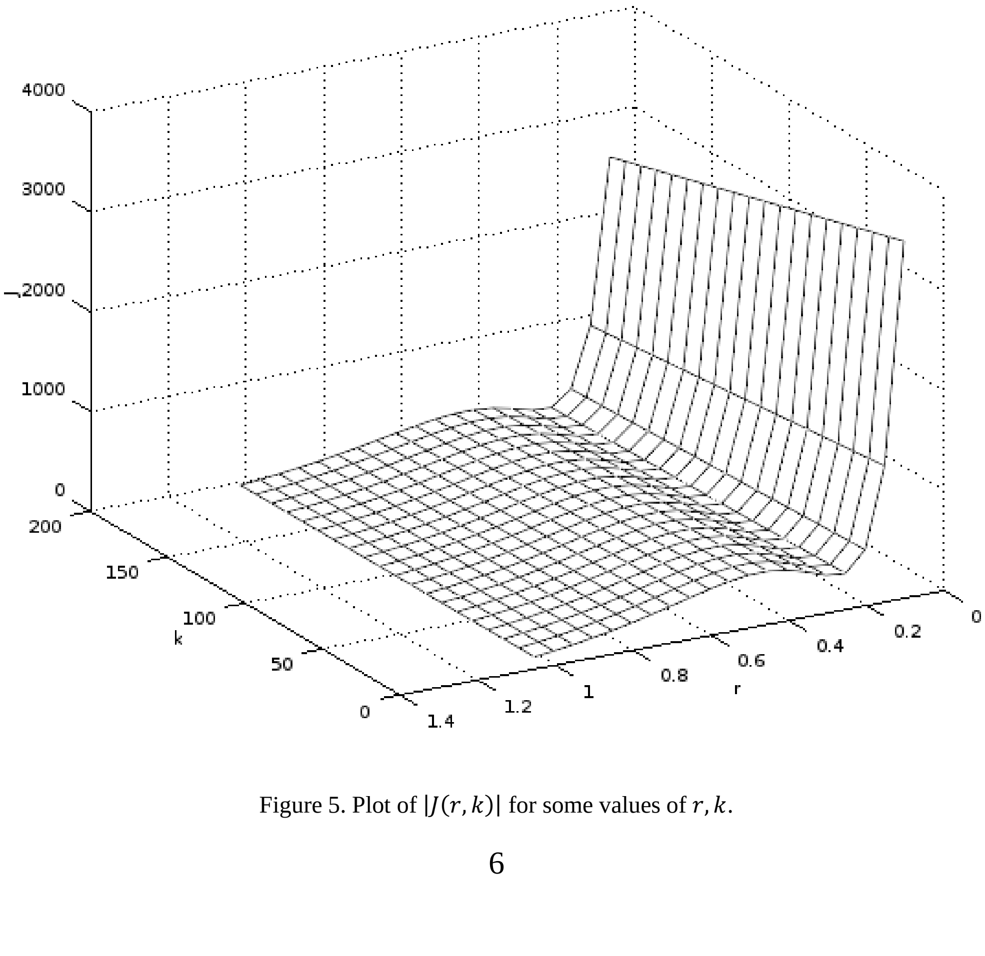

Figure 5. Plot of $|J(r,k)|$ for some values of r, k.

It is naught when $k = f_0$ and for the value of r that represents the solution of the following equation:

Eq. 17 $$\frac{f_2(t_2-t_0)}{f_1(t_1-t_0)} = \frac{\left[f_2\frac{e^{-r(t_2-t_0)}}{f_0}-1\right]e^{(t_2-t_0)}}{\left[f_1\frac{e^{-r(t_1-t_0)}}{f_0}-1\right]e^{-(t_1-t_0)}}$$

Now, within the range of values for r, k, we have that $|J(r,k)| > 0$ (see Figure 5). Then the following iterative method (named newton method) converges (7) to the solution of our system:

Eq. 18 $$\begin{bmatrix} r^{(i+1)} \\ k^{(i+1)} \end{bmatrix} = \begin{bmatrix} r^{(i)} \\ k^{(i)} \end{bmatrix} - \begin{bmatrix} -f_1(t_1-t_0)\left(\frac{k^{(i)}}{f_0}-1\right)e^{-r^{(i)}(t_1-t_0)} & f_1\frac{e^{-r^{(i)}(t_1-t_0)}}{f_0}-1 \\ -f_2(t_2-t_0)\left(\frac{k^{(i)}}{f_0}-1\right)e^{-r^{(i)}(t_2-t_0)} & f_2\frac{e^{-r^{(i)}(t_2-t_0)}}{f_0}-1 \end{bmatrix}^{-1} \begin{bmatrix} f_1\left[1+\left(\frac{k^{(i)}}{f_0}-1\right)e^{-r(t_1-t_0)}\right]-k^{(i)} \\ f_2\left[1+\left(\frac{k^{(i)}}{f_0}-1\right)e^{-r(t_2-t_0)}\right]-k^{(i)} \end{bmatrix}$$

If we use this model for the number of total deaths at time t we have:

Eq. 19 $$d(t) = \frac{k}{1+\left(\frac{k}{d_0}-1\right)e^{-r(t-t_0)}}$$

The software I coded fixes the point (t_0, d_0) with the first measure available and (t_f, d_f) with the last one. Then it fixes the third point with the second experimental measure available and calculates r, k; then it does the same thing with the third experimental measure and calculates r, k; and so forth. In the end, it averages the values for r, k and uses these means to plot the function in Eq. 14. In Figure 6 (left) the logistic curve built with this method using the time series of the number of deaths in Italy from February 21st to March 16th. The three red spots are the numbers of deaths for the following three days, not used for building the curve.
If we use this method for China, we get the plot in Figure 7. As you can see, this cure does not adapt well to the experimental data, and that problem will be addressed with future developments of the model I am already working on. Importantly, this logistic curve has the following parameters: $r = 0.29713, k = 3122.0$. If we average the parameters obtained from the time series of Italy with the ones obtained from China, we get the diagram in Figure 6 (right), which gives a slightly better prevision for the two red dots.

4.1 Script 1

This is the script I use to plot the diagram in Figure 5.

```
% file name = Coronavirus_logistic_jacoban
% date of creation = 08/03/2020
% number of contages in Italy
clear all
% experimental data
f_0 = 750;
t_0 = 8;
f_1 = 2000;
t_1 = 12;
f_2 = 4700;
t_2 = 16;
% initial guess
r(1) = 0.1;
```

```
k(1) = 1.;
N = 20;
for i=1:N
  for j=1:N
     r(i+1) = r(i) + 0.05;
     k(j+1) = k(j) + 10.;
     d_1_r  = -f_1*( -1+ k(j)/f_0 )*(t_1 - t_0)*e^(-r(i)*(t_1-t_0));
     d_1_k  = -1+(f_1/f_0)*e^(-r(i)*(t_1-t_0));
     d_2_r  = -f_2*( -1+ k(j)/f_0 )*(t_2 - t_0)*e^(-r(i)*(t_2-t_0));
     d_2_k  = -1+(f_2/f_0)*e^(-r(i)*(t_2-t_0));
     M(1,1) = d_1_r;
     M(1,2) = d_1_k;
     M(2,1) = d_2_r;
     M(2,2) = d_2_k;
     J(i,j) = abs(det(M));
     M_2=inv(M);
  endfor
endfor
min_J = min ( min ( J(:,:) ) )
figure(1)
colormap ([0,0,0])
mesh(k(1:N), r(1:N), J(1:N,1:N));
grid on
xlabel('k');
ylabel('r');
zlabel('J');
```

4.2 Script 2

This is the script used to plot the diagram in Figure 7.

```
% file name = Coronavirus_logistic_newton_4_deaths_China
% date of creation = 16/03/2020
% nmber of deaths
clear all
% experimental data for deaths from 22/01/2020 for China
f = [17,17,24,40,52,76,125,125,162,204,249,350,414,479,549,618,699,780,871,\
974,1068,1068,1310,1457,1596,1696,1789,1921,2029,2144,2144,2346,2346,2495,\
2563,2615,2641,2682,2727,2761,2803,2835,2871,2902,2931,2959,2986,3008,3024,\
3046,3056,3062,3075,3085,3099,3111,3122];
size = size(f);
days = size(2);
t (1:days) = [1:days]
% we fix f_0 and f_2
d_0 = 5
t_0 = t(d_0);
f_0 = f(d_0);
%
t_2 = t(days);
f_2 = f(days);
% initial guess
r(1) = 0.3;
k(1) = 0.3;
% adjustement
L = 10;              %change of L% is admtted in data to fit a logistic curve
den = 0;             %it counts the number of entry that are considered
stop = 0;            %it has a value of 1 when det(M)=0
% settng some special days
for i = 1:days
  if ( (i+21) == 32 )
```

```
    feb_1 = i;
  endif
  if ( (i+21) == 32+14 )
    feb_15 = i;
  endif
  if ( (i+21) == 32+29 )
    mar_1 = i;
  endif
  if ( (i+21) == 32+29+14 )
    mar_15 = i;
  endif
endfor
% calculation of r, k
for m = d_0+2:days - 1
  f_1 = f(m);
  t_1 = m;
  for l=1:L
    i = 1;
    do
      d_1_r = -f_1*( -1+ k(i)/f_0 )*(t_1 - t_0)*e^(-r(i)*(t_1-t_0));
      d_1_k = -1+(f_1/f_0)*e^(-r(i)*(t_1 - t_0));
      d_2_r = -f_2*( -1+ k(i)/f_0 )*(t_2 - t_0)*e^(-r(i)*(t_2-t_0));
      d_2_k = -1+(f_2/f_0)*e^(-r(i)*(t_2 - t_0));
      M(1,1) = d_1_r;
      M(1,2) = d_1_k;
      M(2,1) = d_2_r;
      M(2,2) = d_2_k;
      if (abs(det(M))>10^(-24))
        M_2=inv(M);
        psi_1 = f_1*( 1 + ((k(i)/f_0) - 1)*e^(-r(i)*(t_1 - t_0)) )-k(i);
        psi_2 = f_2*( 1 + ((k(i)/f_0) - 1)*e^(-r(i)*(t_2 - t_0)) )-k(i);
        sol = [r(i), k(i)]' - M_2*[psi_1, psi_2]';
        r(i+1) = sol (1,1);
        k(i+1) = sol (2,1);
      else
          r(i+1) = r (i);
          k(i+1) = k(i);
          stop = 1;
      endif
      i = i+1;
    until (  (( k(i) - k(i-1) )^2 + ( r(i) - r(i-1) )^2)<0.01  )
    r_s (m) = r(i);
    k_s (m) = k(i);
    if ( (r_s (m) < 0) | (k_s (m) < 0) | (stop==1) )
      r_s (m) = 0.;
      k_s (m) = 0.;
      f_1 = f_1 + (f_1)/100;
      stop=0;
    else
      den = den + 1;
      break
    endif
  endfor
  if ( (r_s < 0) | (k_s < 0) )
    f_1 = f(m);
    for l=1:L
      i=1;
      do
        d_1_r = -f_1*( -1+ k(i)/f_0 )*(t_1 - t_0)*e^(-r(i)*(t_1-t_0));
        d_1_k = -1+(f_1/f_0)*e^(-r(i)*(t_1 - t_0));
        d_2_r = -f_2*( -1+ k(i)/f_0 )*(t_2 - t_0)*e^(-r(i)*(t_2-t_0));
```

```
        d_2_k = -1+(f_2/f_0)*e^(-r(i)*(t_2 - t_0));
        M(1,1) = d_1_r;
        M(1,2) = d_1_k;
        M(2,1) = d_2_r;
        M(2,2) = d_2_k;
        if (abs(det(M))>10^(-24))
          M_2=inv(M);
          psi_1 = f_1*( 1 + ((k(i)/f_0) - 1)*e^(-r(i)*(t_1 - t_0)) )-k(i);
          psi_2 = f_2*( 1 + ((k(i)/f_0) - 1)*e^(-r(i)*(t_2 - t_0)) )-k(i);
          sol = [r(i), k(i)]' - M_2*[psi_1, psi_2]';
          r(i+1) = sol (1,1);
          k(i+1) = sol (2,1);
        else
          r(i+1) = r (i);
          k(i+1) = k(i);
          stop = 1;
        endif
        i = i+1;
      until (  (( k(i) - k(i-1) )^2 + ( r(i) - r(i-1) )^2)<0.01  )
      r_s(m) = r(i);
      k_s(m) = k(i);
      if ( (r_s(m) < 0)|(k_s(m) < 0)|(stop==1) )
        stop = 0;
        r_s (m) = 0.;
        k_s (m) = 0.;
        f_1 = f_1 - (f_1)/100;
      else
        den = den + 1;
        break
      endif
    endfor
  endif
endfor
% it calculates the means
r_s_m = 0;
k_s_m = 0;
for p =d_0+2:days-1
  r_s_m = r_s_m + r_s(p);
  k_s_m = k_s_m + k_s(p);
endfor
r_s_m = r_s_m/den
k_s_m = k_s_m/den
N = 0                          % days for which the simulation is done
% plotting
for h=d_0:days+N
  t (h) = h;
  f_s(h) = k_s_m/( 1 + ( (k_s_m/f_0)  - 1)*( e^( -r_s_m*(t(h) - t_0) ) ) );
endfor
% plotting
for h=d_0:days+N
  t (h) = h;
  f_s(h) = k_s_m/( 1 + ( (k_s_m/f_0)  - 1)*( e^( -r_s_m*(t(h) - t_0) ) ) );
endfor
for h=d_0:days+N
  t (h) = h;
  f_s(h) = k_s_m/( 1 + ( (k_s_m/f_0)  - 1)*( e^( -r_s_m*(t(h) - t_0) ) ) );
endfor
figure (1)
plot (t (d_0:days+N), f_s(d_0:days+N), '-k', "linewidth", 2)
xlabel('days after February 21');
ylabel('number of deaths');
```

```
axis ([0, days+N, -100, max(f_s(:))+1000])
grid on
text (feb_1, 300,'Feb 01')
text (feb_15, 300,'Feb 15')
text (mar_1, 300, 'Mar 01')
text (mar_15, 300, 'Mar 15')
hold on
plot ([feb_1,feb_1], [-100, f_s(feb_1)], ':k', 'linewidth', 2)
plot ([feb_15,feb_15], [-100, f_s(feb_15)], ':k', 'linewidth', 2)
plot ([mar_1,mar_1], [-100,f_s(mar_1)], ':k', 'linewidth', 2)
plot ([mar_15,mar_15], [-100,f_s(mar_15)], ':k', 'linewidth', 2)
% plotting empirical data
for g=1:days
  plot (g,f(g),'@k.','markersize',10)
endfor
```

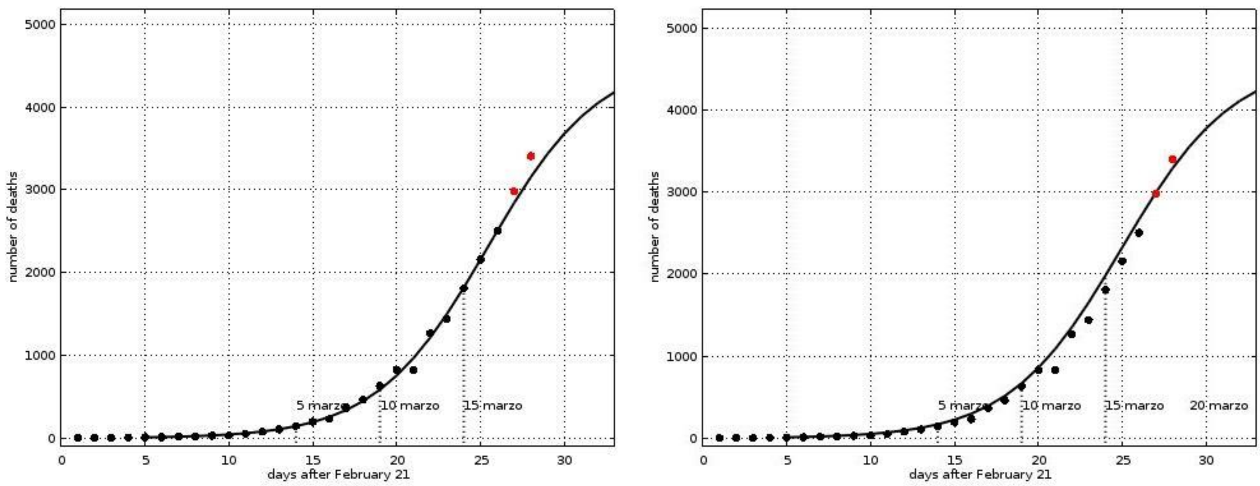

Figure 6. *Left.* The logistic curve for the number of deaths in Italy. Black spots are the measures used for building the curve, the red spots are the actual numbers of deaths for the following 3 days. *Right.* This curve is obtained averaging the values r, k with the ones calculated for China.

5 Number of infected people

If we indicate $c(t)$ the number of infected individuals at time *t*, and with $d(t)$ the number of deaths we have

Eq. 20 $$c(t) = \frac{d(t+\delta)}{m/100}$$

We are interested in the distribution of C_t, the random variable that represents the number of infected individuals at time *t*. If we define the random variable $Z = 1/M$ and if f_Z is its probability density, we can say that the conditional density of C_t given $\Delta = \delta$ is

Eq. 21 $$\bar{f}_{C_t|\Delta}(c(t)|\delta) = \frac{1}{\frac{d(t+\delta)}{1/100}} f_Z\left(\frac{c(t)}{\frac{d(t+\delta)}{1/100}}\right) = \frac{1}{10^2 d(t+\delta)} f_Z\left(\frac{1}{10^2 d(t+\delta)} c(t)\right)$$

But it can also be proven (5) that

Eq. 22 $$f_Z(z) = \begin{cases} \frac{\lambda_M^{\alpha_M}}{\Gamma(\alpha_M)} z^{-\alpha_M - 1} e^{-\frac{\lambda}{z}} & \forall z > 0 \\ 0 & \forall z \leq 0 \end{cases}$$

which means that we have found

$$\bar{f}_{C_t|\Delta}(c(t)|\delta) = \frac{\lambda_M^{\alpha_M}}{10^2 d(t+\delta)\Gamma(\alpha_M)} \left(\frac{c(t)}{10^2 d(t+\delta)}\right)^{-(\alpha_M+1)} e^{-\frac{\lambda_M d(t+\delta)}{c(t)} 10^2} \Rightarrow$$

Eq. 23 $$\bar{f}_{C_t|\Delta}(c(t)|\delta) = \frac{\lambda_M^{\alpha_M}}{\Gamma(\alpha_M)} \frac{\left(10^2 d(t+\delta)\right)^{\alpha_M}}{c(t)^{(\alpha_M+1)}} e^{-\frac{\lambda_M d(t+\delta)}{c(t)} 10^2}$$

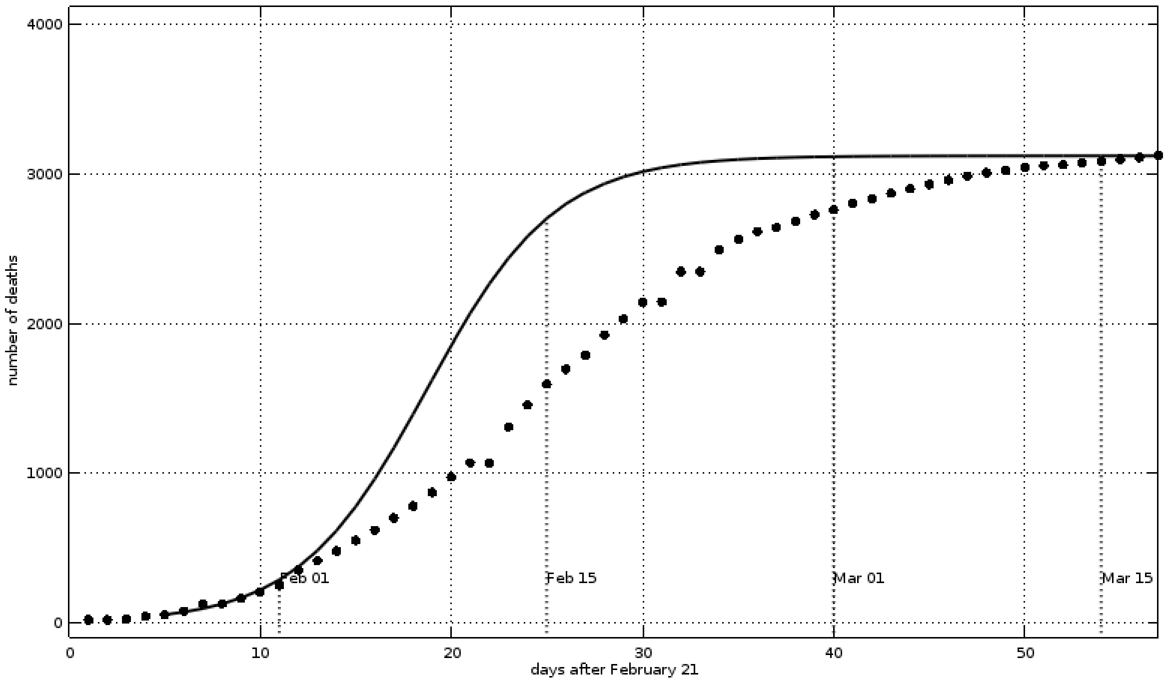

Figure 7. The logistic curve that best fits the time series of deaths in China. We get: $r = 0.29713, k = 3122.0$.

If we now introduce the random vector $\boldsymbol{X} = [C_t, \Delta]$, we can write

Eq. 24 $$\bar{f}_{C_t|\Delta}(c(t)|\delta) = \frac{f_{\boldsymbol{X}}(c(t),\delta)}{f_\Delta(\delta)}$$

where $f_{\boldsymbol{X}}(c(t),\delta)$ is the joint density of the probability of $[C_t, \Delta]$. This means that:

$$\frac{f_{\boldsymbol{X}}(c(t),\delta)}{f_\Delta(\delta)} = \frac{\lambda_M^{\alpha_M}}{\Gamma(\alpha_M)} \frac{\left(10^2 d(t+\delta)\right)^{\alpha_M}}{c(t)^{(\alpha_M+1)}} e^{-\frac{\lambda_M d(t+\delta)}{c(t)} 10^2} \Rightarrow$$

Eq. 25 $$f_{\boldsymbol{X}}(c(t),\delta) = \frac{\lambda_M^{\alpha_M}}{\Gamma(\alpha_M)} \frac{\left(10^2 d(t+\delta)\right)^{\alpha_M}}{c(t)^{(\alpha_M+1)}} e^{-\frac{\lambda_M d(t+\delta)}{c(t)} 10^2} \frac{\lambda_\Delta^{\alpha_\Delta}}{\Gamma(\alpha_\Delta)} \delta^{(\alpha_\Delta - 1)} e^{-\lambda_\Delta \delta} \quad \forall \delta, c(t) > 0$$

where we have considered the expression of $f_\Delta(\delta)$ in Eq. 6. The marginal probability density of $c(t)$ can then be calculated as follows:

Eq. 26 $$f_{C_t}(c(t)) = \int_0^\infty f_{\boldsymbol{X}}(c(t),\delta)\, d\delta$$

Now we can calculate the marginal density of probability for C_t:

$$f_{C_t}(c(t)) = \int_0^{\infty} f_X(c(t), \delta)\, d\delta \Rightarrow$$

Eq. 27 $$f_{C_t}(c(t)) = \frac{\lambda_M^{\alpha_M} \lambda_\Delta^{\alpha_\Delta}}{\Gamma(\alpha_M)\Gamma(\alpha_\Delta) c(t)^{(\alpha_M+1)}} \int_0^{\infty} \frac{[d(t+\delta)]^{\alpha_M} \delta^{(\alpha_\Delta - 1)}}{e^{\left(\frac{\lambda_M d(t+\delta)}{c(t)} + \lambda_\Delta \delta\right)}} d\delta$$

We can then calculate the expectation and the standard deviation of this probability density by using the following formulae:

Eq. 28 $$E[c(t)] = \int_0^{+\infty} \zeta f_{C_t}(\zeta) d\zeta$$

Eq. 29 $$\sigma_{C_t}^2 = Var[c(t)] = \int_0^{+\infty} (\zeta - E[c(t)])^2 f_{C_t}(\zeta) d\zeta$$

5.1 Numeric method

Instead of using the probability density in Eq. 27 we can calculate the distribution of C_t considering that the probability associated with the value $c(t) = \frac{d(t+\delta)}{m/100}$ can be approximately expressed as

Eq. 30 $$P\left(C_t = c(t) = \frac{d(t+\delta)}{m/100}\right) = 1 \cdot f_{C_t}\big(c(t)\big) = 1 \cdot f_M(m) \cdot 1 \cdot f_\Delta(\delta)$$

Then we have that the discrete distribution of $c(t)$ is given by the following set

Eq. 31 $$\left\{c(t) = \frac{d(t+\delta)}{m/100} f_M(m) f_\Delta(\delta) | m \in \{0,1,2,\dots,13\}\ and\ \delta \in \{0,1,\dots,45\}\right\}$$

Then we can easily calculate the mean of this set. We can also find the range of values for $c(t)$ that includes a probability of, for instance, 50%. In Figure 8.

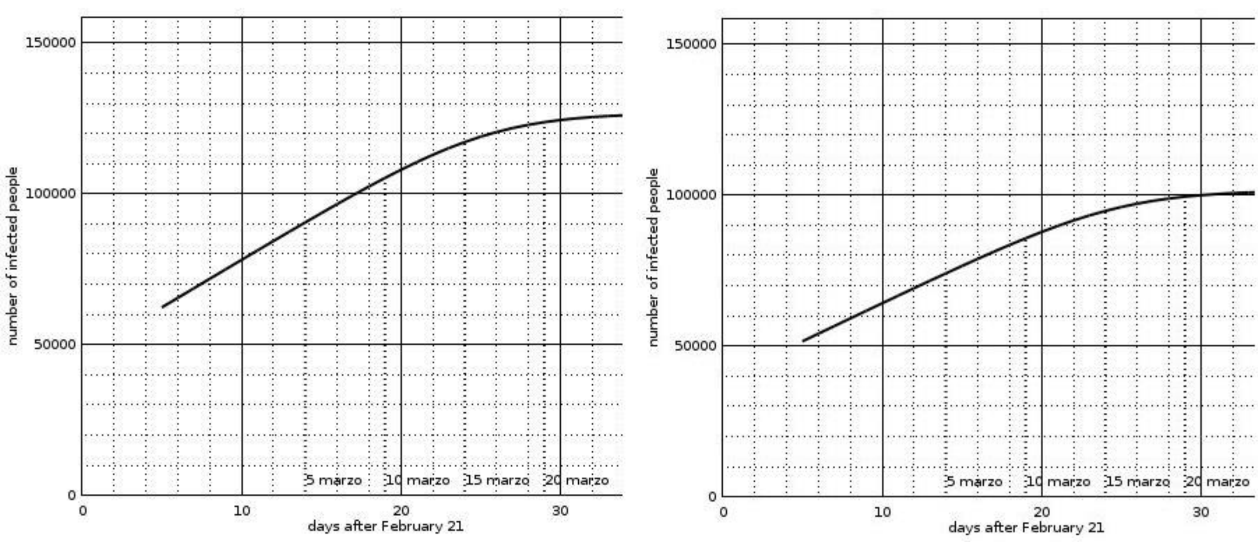

Figure 8. *Left.* The mean value for $c(t)$ calculated using only the time series for deaths for Italy. *Right.* The mean value for $c(t)$ calculated averaging r, k obtained from the time series for deaths for Italy with the ones obtained from the same time series from China.

5.2 Script

This is the script described in the present paper.

```
% file name = Coronavirus_logistic_newton_4_continous
% date of creation = 16/03/2020
% number of contages in Italy
clear all
mu_M = 7;
mu_delta = 16;                              % days
sigma_delta = 8.81;                          % days
lambda_delta = mu_delta/(sigma_delta^2);       % 1/days
alpha_delta = (mu_delta^2)/(sigma_delta^2);
gamma_delta = gamma (alpha_delta)
%it searches for lambda_M, alpha_M, sigma_M
delta = 18/1000.
for i = 1:100
  sigma_M = 0.1*i;
  lambda_M = mu_M/(sigma_M^2);
  alpha_M = (mu_M^2)/(sigma_M^2);
  gamma_M = gamma (alpha_M);
  for k=222:555
    m(k)= delta*(k-1);
    f_M(k) = ( ( lambda_M^alpha_M )/( gamma_M ) )*( m(k)^( alpha_M - 1. ) )*( e^( (-1)*lambda_M*m(k) ) );
  endfor
  k = 222;
  F_M(k) = f_M(k);
  F_M(k+2) = F_M(k) + delta*( f_M(k) + ( 4*f_M(k+1) ) + f_M(k+2) )/3.;
  F_M(k+1) = F_M(k+2)*0.5;
  for k=223:553
    F_M(k+2) = F_M(k) + delta*( f_M(k) + ( 4*f_M(k+1) ) + f_M(k+2) )/3.;
  end
  P_M(i) = F_M(k+2);
 endfor
temp_M = min( abs(P_M(1:100)-0.95) );
for i=1:100
  if ( abs(P_M(i)-0.95) == temp_M )
    sigma_M = 0.1*i
    lambda_M = mu_M/(sigma_M^2)
    alpha_M = (mu_M^2)/(sigma_M^2)
    gamma_M = gamma (alpha_M)
    break
  endif
endfor
% it plots the density of mortality
delta = 18/1000.
for k=1:1:1000
  m(k)= delta*(k-1);
  f_M(k) = ( ( lambda_M^alpha_M )/( gamma_M ) )*( m(k)^( alpha_M - 1. ) )*( e^( (-1)*lambda_M*m(k) ) );
endfor
F_M(1) = 0.;
F_M(3) = F_M(1) + delta*( f_M(1) + ( 4*f_M(2) ) + f_M(3) )/3.;
F_M(2) = F_M(3)*0.5;
for k=2:1:998
  F_M(k+2) = F_M(k) + delta*( f_M(k) + ( 4*f_M(k+1) ) + f_M(k+2) )/3.;
end
figure (1)
plot(m(:),f_M(:),'-k','Linewidth', 3)
xlabel('m')
ylabel('{f_M}(m)')
```

```
grid minor
figure (2)
plot(m(:),F_M(:),'-k','Linewidth', 3)
xlabel('m')
ylabel('{F_M}(m)')
grid minor
% it plots the density of delta
delta = 45./1000.
for k=1:1:1000
  d(k)= delta*(k-1);
  P_delta(k) = ( ( lambda_delta^alpha_delta )/( gamma_delta ) )*\
  ( d(k)^( alpha_delta - 1. ) )*( e^( (-1)*lambda_delta*d(k) ) );
endfor
F_delta(1) = 0.;
F_delta(3) = F_delta(1) + delta*( P_delta(1) + ( 4*P_delta(2) ) + P_delta(3) )/3.;
F_delta(2) = F_delta(3)*0.5;
for k=2:1:998
  F_delta(k+2) = F_delta(k) + delta*( P_delta(k) + ( 4*P_delta(k+1) ) + P_delta(k+2) )/3.;
end
figure (3)
plot(d(:),P_delta(:),'-k','Linewidth', 2)
xlabel('{\delta}')
ylabel('f_{\Delta}({\delta})')
grid minor
axis ([0,45,0,0.06])
figure (4)
plot(d(:),F_delta(:),'-k','Linewidth', 2)
xlabel('{\delta}')
ylabel('F_{\Delta}({\delta})')
grid minor
axis ([0,45,0,1.20])
% this is the section in which it is calculated the logistic curve for the number of deaths
% experimental data for deaths in Italy from Feb 21
f = [1,2,3,7,10,12,17,21,29,34,52,79,107,148,197,233,366,\
463,631,827,827,1266,1441,1809,2158,2503,2978,3405]
size = size(f);
days = size(2);
t (1:days) = [1:days]
% we fix f_0 and f_2
d_0 = 5
t_0 = t(d_0);
f_0 = f(d_0);
t_2 = t(days);
f_2 = f(days);
% initial guess
r(1) = 0.3;
k(1) = 0.3;
% adjustement
L = 10;             %change of L% is admtted in data to fit a logistic curve
den = 0;            %it counts the number of entry that are considered
stop = 0;           %it has a value of 1 when det(M)=0
% calculation of r, k
for m = d_0+2:days - 1
  f_1 = f(m);
  t_1 = m;
  for l=1:L
    i = 1;
    do
      d_1_r = -f_1*( -1+ k(i)/f_0 )*(t_1 - t_0)*e^(-r(i)*(t_1-t_0));
      d_1_k = -1+(f_1/f_0)*e^(-r(i)*(t_1 - t_0));
      d_2_r = -f_2*( -1+ k(i)/f_0 )*(t_2 - t_0)*e^(-r(i)*(t_2-t_0));
```

```
    d_2_k = -1+(f_2/f_0)*e^(-r(i)*(t_2 - t_0));
    M(1,1) = d_1_r;
    M(1,2) = d_1_k;
    M(2,1) = d_2_r;
    M(2,2) = d_2_k;
    if (abs(det(M))>10^(-24))
      M_2=inv(M);
      psi_1 = f_1*( 1 + ((k(i)/f_0) - 1)*e^(-r(i)*(t_1 - t_0)) )-k(i);
      psi_2 = f_2*( 1 + ((k(i)/f_0) - 1)*e^(-r(i)*(t_2 - t_0)) )-k(i);
      sol = [r(i), k(i)]' - M_2*[psi_1, psi_2]';
      r(i+1) = sol (1,1);
      k(i+1) = sol (2,1);
    else
        r(i+1) = r (i);
        k(i+1) = k(i);
        stop = 1;
    endif
    i = i+1;
  until (  (( k(i) - k(i-1) )^2 + ( r(i) - r(i-1) )^2)<0.01  )
  r_s (m) = r(i);
  k_s (m) = k(i);
  if ( (r_s (m) < 0) | (k_s (m) < 0) | (stop==1) )
    r_s (m) = 0.;
    k_s (m) = 0.;
    f_1 = f_1 + (f_1)/100;
    stop=0;
  else
    den = den + 1;
    break
  endif
endfor
if ( (r_s < 0) | (k_s < 0) )
  f_1 = f(m);
  for l=1:L
    i=1;
    do
      d_1_r = -f_1*( -1+ k(i)/f_0 )*(t_1 - t_0)*e^(-r(i)*(t_1-t_0));
      d_1_k = -1+(f_1/f_0)*e^(-r(i)*(t_1 - t_0));
      d_2_r = -f_2*( -1+ k(i)/f_0 )*(t_2 - t_0)*e^(-r(i)*(t_2-t_0));
      d_2_k = -1+(f_2/f_0)*e^(-r(i)*(t_2 - t_0));
      M(1,1) = d_1_r;
      M(1,2) = d_1_k;
      M(2,1) = d_2_r;
      M(2,2) = d_2_k;
      if (abs(det(M))>10^(-24))
        M_2=inv(M);
        psi_1 = f_1*( 1 + ((k(i)/f_0) - 1)*e^(-r(i)*(t_1 - t_0)) )-k(i);
        psi_2 = f_2*( 1 + ((k(i)/f_0) - 1)*e^(-r(i)*(t_2 - t_0)) )-k(i);
        sol = [r(i), k(i)]' - M_2*[psi_1, psi_2]';
        r(i+1) = sol (1,1);
        k(i+1) = sol (2,1);
      else
        r(i+1) = r (i);
        k(i+1) = k(i);
        stop = 1;
      endif
      i = i+1;
    until (  (( k(i) - k(i-1) )^2 + ( r(i) - r(i-1) )^2)<0.01  )
    r_s(m) = r(i);
    k_s(m) = k(i);
    if ( (r_s(m) < 0)|(k_s(m) < 0)|(stop==1) )
```

```
        stop = 0;
        r_s (m) = 0.;
        k_s (m) = 0.;
        f_1 = f_1 - (f_1)/100;
      else
        den = den + 1;
        break
      endif
    endfor
  endif
endfor
% it calculates the means
r_s_m = 0;
k_s_m = 0;
for p =d_0+2:days-1
  r_s_m = r_s_m + r_s(p);
  k_s_m = k_s_m + k_s(p);
endfor
r_s_m = r_s_m/den
k_s_m = k_s_m/den
N = 50                          % days for which the simulation is done
% averaging r and k with values from China
r_s_m_c = (r_s_m+0.29713)/2
k_s_m_c = (k_s_m+3122.0)/2
% plotting
for h=d_0:days+N
  t (h) = h;
  f_s(h) = k_s_m/( 1 + ( (k_s_m/f_0)  - 1)*( e^( -r_s_m*(t(h) - t_0) ) ) );
endfor
figure (5)
plot (t (d_0:days+N), f_s(d_0:days+N), '-k', "linewidth", 2)
xlabel('days after February 21');
ylabel('number of deaths');
axis ([0, days+N, -100, max(f_s(:))+1000])
grid on
text (14, 400,'5 marzo')
text (19, 400,'10 marzo')
text (24, 400, '15 marzo')
text (29, 400,'20 marzo')
text (34, 400, '25 marzo')
hold on
plot ([14,14], [-100, f_s(14)], ':k', 'linewidth', 2)
plot ([19,19], [-100, f_s(19)], ':k', 'linewidth', 2)
plot ([24,24], [-100,f_s(24)], ':k', 'linewidth', 2)
plot ([29,29], [-100,f_s(29)], ':k', 'linewidth', 2)
plot ([34,34], [-100,f_s(34)], ':k', 'linewidth', 2)
% plotting empirical data
for g=1:days
  plot (g,f(g),'@k.','markersize',10)
endfor
% This is the averaged curve with data from China
for h=d_0:days+N
  t (h) = h;
  f_s(h) = k_s_m/( 1 + ( (k_s_m_c/f_0)  - 1)*( e^( -r_s_m_c*(t(h) - t_0) ) ) );
endfor
figure (6)
plot (t (d_0:days+N), f_s(d_0:days+N), '-k', "linewidth", 2)
xlabel('days after February 21');
ylabel('number of deaths');
axis ([0, days+N, -100, max(f_s(:))+1000])
grid on
```

```
text (14, 400,'5 marzo')
text (19, 400,'10 marzo')
text (24, 400, '15 marzo')
text (29, 400,'20 marzo')
text (34, 400, '25 marzo')
hold on
plot ([14,14], [-100, f_s(14)], ':k', 'linewidth', 2)
plot ([19,19], [-100, f_s(19)], ':k', 'linewidth', 2)
plot ([24,24], [-100,f_s(24)], ':k', 'linewidth', 2)
plot ([29,29], [-100,f_s(29)], ':k', 'linewidth', 2)
plot ([34,34], [-100,f_s(34)], ':k', 'linewidth', 2)
% plotting empirical data
for g=1:days
  plot (g,f(g),'@k.','markersize',10)
endfor
% distribution for the number of infected people
n_M = 13;
n_delta = 40;
d_fin = days + N - n_delta
for j = d_0:d_fin
  i = 0;                          % this is the number of values for c(j)
  c_mean (j) = 100*(k_s_m/mu_M)/( 1 + ( (k_s_m/f_0)  - 1)*( e^( -r_s_m*(t(j) + mu_delta - t_0) ) ) );
  for m = 1:n_M
    for delta = 1:n_delta
      i = i+1;
      c(i, j) = 100*(k_s_m/m)/( 1 + ( (k_s_m/f_0)  - 1)*( e^( -r_s_m*(t(j) + delta - t_0) ) ) );
    endfor
  endfor
  mu_C (j) = mean ( c(1:i, j) );
  sigma_C (j) = std ( c(1:i, j) );
  sup_C (j) = mu_C (j) + (sigma_C (j)/4);
  inf_C (j) = mu_C (j) - (sigma_C (j)/4);
endfor
% probability density
for j = d_0:d_fin
  i = 0;                          % this is the number of values for c(j)
  P_C (j) = 0;
  for m = 1:n_M
    for delta = 1:n_delta
      i = i+1;
      if ( (c(i, j)<sup_C (j)) & (c(i, j)>inf_C (j)) )
        P_C (j) = P_C (j) + ( P_M(m)*P_delta(delta) );
      endif
    endfor
  endfor
endfor
% this is the calculus using r_m_s_c and k_m_s_c
for j = d_0:d_fin
  i = 0;                          % this is the number of values for c(j)
  c_mean_c (j) = 100*(k_s_m_c/mu_M)/( 1 + ( (k_s_m_c/f_0)  - 1)*( e^( -r_s_m_c*(t(j) + mu_delta - t_0) ) )
);
  for m = 1:n_M
    for delta = 1:n_delta
      i = i+1;
      c_c(i, j) = 100*(k_s_m_c/m)/( 1 + ( (k_s_m_c/f_0)  - 1)*( e^( -r_s_m_c*(t(j) + delta - t_0) ) ) );
    endfor
  endfor
  mu_C_c (j) = mean ( c_c(1:i, j) );
  sigma_C_c (j) = std ( c_c(1:i, j) );
  sup_C_c (j) = mu_C_c (j) + (sigma_C_c (j)/4);
  inf_C_c (j) = mu_C_c (j) - (sigma_C_c (j)/4);
```

```
endfor
% plotting
figure (7)
plot (t (d_0:d_fin), mu_C(d_0:d_fin), '-k', "linewidth", 2)
hold on
xlabel('days after February 21');
ylabel('number of infected people');
grid on
grid minor
text (14, 5000,'5 marzo')
text (19, 5000,'10 marzo')
text (24, 5000, '15 marzo')
text (29, 5000,'20 marzo')
hold on
plot ([14,14], [-100, mu_C(14)], ':k', 'linewidth', 2)
plot ([19,19], [-100, mu_C(19)], ':k', 'linewidth', 2)
plot ([24,24], [-100, mu_C(24)], ':k', 'linewidth', 2)
plot ([29,29], [-100, mu_C(29)], ':k', 'linewidth', 2)
axis ([0, d_fin, 0, max(sup_C(:))])
figure (8)
plot (t (d_0:d_fin), P_C(d_0:d_fin), '-k', "linewidth", 2)
xlabel('days after February 21');
ylabel('probability to be within the range');
grid on
text (14, 0.05,'5 marzo')
text (19, 0.05,'10 marzo')
text (24, 0.05, '15 marzo')
text (29, 0.05,'20 marzo')
hold on
plot ([14,14], [0, P_C(14)], ':k', 'linewidth', 2)
plot ([19,19], [0, P_C(19)], ':k', 'linewidth', 2)
plot ([24,24], [0, P_C(24)], ':k', 'linewidth', 2)
plot ([29,29], [0, P_C(29)], ':k', 'linewidth', 2)
figure (9)
plot (t (d_0:d_fin), mu_C_c (d_0:d_fin), '-k', "linewidth", 2)
xlabel('days after February 21');
ylabel('number of infected people');
grid on
text (14, 5000,'5 marzo')
text (19, 5000,'10 marzo')
text (24, 5000, '15 marzo')
text (29, 5000,'20 marzo')
hold on
plot ([14,14], [-100, mu_C_c(14)], ':k', 'linewidth', 2)
plot ([19,19], [-100, mu_C_c(19)], ':k', 'linewidth', 2)
plot ([24,24], [-100, mu_C_c(24)], ':k', 'linewidth', 2)
plot ([29,29], [-100, mu_C_c(29)], ':k', 'linewidth', 2)
axis ([0, d_fin, 0, max(sup_C(:))])
grid on
grid minor
```

6 Riferimenti

1. Sun P, Lu X, Xu C, Sun W, Pan B. Understanding of COVID-19 based on current evidence. J Med Virol. 2020.
2. Zhu N, Zhang D, Wang W, Li X, Yang B, Song J, et al. A Novel Coronavirus from Patients with Pneumonia in China, 2019. N Engl J Med. 2020;382(8):727-33.
3. Ji W, Wang W, Zhao X, Zai J, Li X. Cross-species transmission of the newly identified coronavirus 2019-nCoV. J Med Virol. 2020;92(4):433-40.
4. Li LQ, Huang T, Wang YQ, Wang ZP, Liang Y, Huang TB, et al. 2019 novel coronavirus patients' clinical characteristics, discharge rate and fatality rate of meta-analysis. J Med Virol. 2020.
5. Maccallini P. Variabili Aleatorie. Introduzione al calcolo delle probabilità2018.
6. Giaquinta M, Modica G. Analisi Matematica 1. Funzioni di una variabile. Pitagora, editor1998.
7. Gori L. Calcolo numerico. Kappa E, editor1999.
8. Harris M, Taylor G. Medical Statistics Made Easy. Second Ed. ed2008.
9. Ross SM. Introduction to probability and Statistics for Engeneers and Scientists. Fourth Edition ed2009.
10. Wang W, Tang J, Wei F. Updated understanding of the outbreak of 2019 novel coronavirus (2019-nCoV) in Wuhan, China. J Med Virol. 2020;92(4):441-7.