Example 1: Fitting the parameters to the model using simulated data

Table of Contents

A generalized SEIR model is used to simulate an epidemic breakout. Seven different states are considered in the following, as proposed by ref. [1]:

1. Susceptibles cases S(t)
2. Insusceptibles cases P(t)
3. Exposed cases E(t)
4. Infectious cases I(t)
5. Quarantined cases Q(t)
6. Recovered cases R(t)
7. Dead cases D(t)

The parameters are as follows:

- alpha: protection rate
- beta: infection rate
- gamma: Inverse of the average latent time
- delta: rate at which infectious people enter in quarantine
- lambda0 and lambda1: coefficients used in the time-dependant cure rate
- kappa0 and kappa1: coefficient used in the time-dependant mortality rate

The population is assumed constant, i.e. the births and natural death are not modelled. The cure rate and death rate are here time-dependant but they need some empirical coefficients to tune the time-dependency of these parameters.

The generalized SEIR model [1] is:

$$\frac{\mathrm{d}S(t)}{\mathrm{d}t} = -\alpha S(t) - \beta \frac{S(t)I(t)}{N_{pop}}$$

$$\frac{\mathrm{d}E(t)}{\mathrm{d}t} = -\gamma E(t) + \beta \frac{S(t)I(t)}{N_{pop}}$$

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = \gamma E(t) - \delta I(t)$$

$$\frac{\mathrm{d}Q(t)}{\mathrm{d}t} = \delta I(t) - \lambda(t)Q(t) - \kappa(t)Q(t)$$

$$\frac{\mathrm{d}R(t)}{\mathrm{d}t} = \lambda(t)Q(t)$$

$$\frac{\mathrm{d}D(t)}{\mathrm{d}t} = \kappa(t)Q(t)$$

$$\frac{\mathrm{d}P(t)}{\mathrm{d}t} = \alpha S(t)$$

The content presented hereafter is **not** proposed by [1]. Therefore, it can significantly differ from their work. Firstly, I decided to model $\kappa(t)$ and $\lambda(t)$ as:

$$\kappa(t) = \kappa_0 \exp(-\kappa_1 t)$$

$$\lambda(t) = \lambda_0 [1 - \exp(-\lambda_1 t)]$$

The idea behind these functions is that the death rate should become close to zero at time increases while the recovery rate converge toward a constant value.

I also decided to re-write the system of ODEs in a matrix form:

$$\frac{\mathrm{dY}}{\mathrm{dt}} = A * Y + F$$

where

$$Y = [S, E, I, Q, R, D, P]^{\top}$$

$$A = \begin{bmatrix} -\alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\gamma & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma & -\delta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \delta & -\kappa(t) - \lambda(t) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda(t) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \kappa(t) & 0 & 0 & 0 & 0 \\ \alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$F = S(t) \cdot I(t) \cdot \begin{bmatrix} -\frac{\beta}{N_{\mathrm{pop}}} \\ \frac{\beta}{N_{\mathrm{pop}}} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The equation $\frac{\mathrm{dY}}{\mathrm{dt}} = A * Y + F$ is then solved using the classic 4th order Runge-Kutta method.

References

[1] https://arxiv.org/pdf/2002.06563.pdf

Initialisation

Case of an imaginary epidemy outbreak that took place on 2010-01-01. The simulation time is set to 9 months.

```
clearvars;close all;clc;

% Time definition
dt = 0.1; % time step
time1 = datetime(2010,01,01,0,0,0):dt:datetime(2010,09,01,0,0,0);
N = numel(time1);
t = [0:N-1].*dt;
```

Generate the data

```
Npop= 60e6; % population (60 millions)
Q0 = 200; % Initial number of infectious that have bee quanrantined
I0 = Q0; % Initial number of infectious cases non-quarantined
E0 = 0; % Initial number of exposed
R0 = 0; % Initial number of recovereds
D0 = 1; % Initial number of deads
alpha = 0.08; %  protection rate
beta = 0.9; %  infection rate
gamma= 1/2; % inverse of average latent time
delta= 1/8; % rate at which infectious people enter in quarantine
Lambda01 = [0.03 0.05]; % cure rate (time dependant)
Kappa01 =  [0.03 0.05]; % mortality rate (time dependant)
[S,E,I,Q,R,D,P] = SEIQRDP(alpha,beta,gamma,delta,Lambda01,Kappa01,Npop,E0,I0,Q0,R0,D0,t
```

Fit the data

The fitting is done using the time histories of the number of quarantined $Q(t)$, recovered $R(t)$ and deads $D(t)$ only. The number of exposed, susceptible, insusceptible and infectious is computed in the model but not used as target.

```
guess = [0.05,0.9,1/4,1/10,0.03,0.03,0.02,0.06]; % initial guess
[alpha1,beta1,gamma1,delta1,Lambda1,Kappa1] = fit_SEIQRDP(Q,R,D,Npop,E0,I0,time1,guess)
```

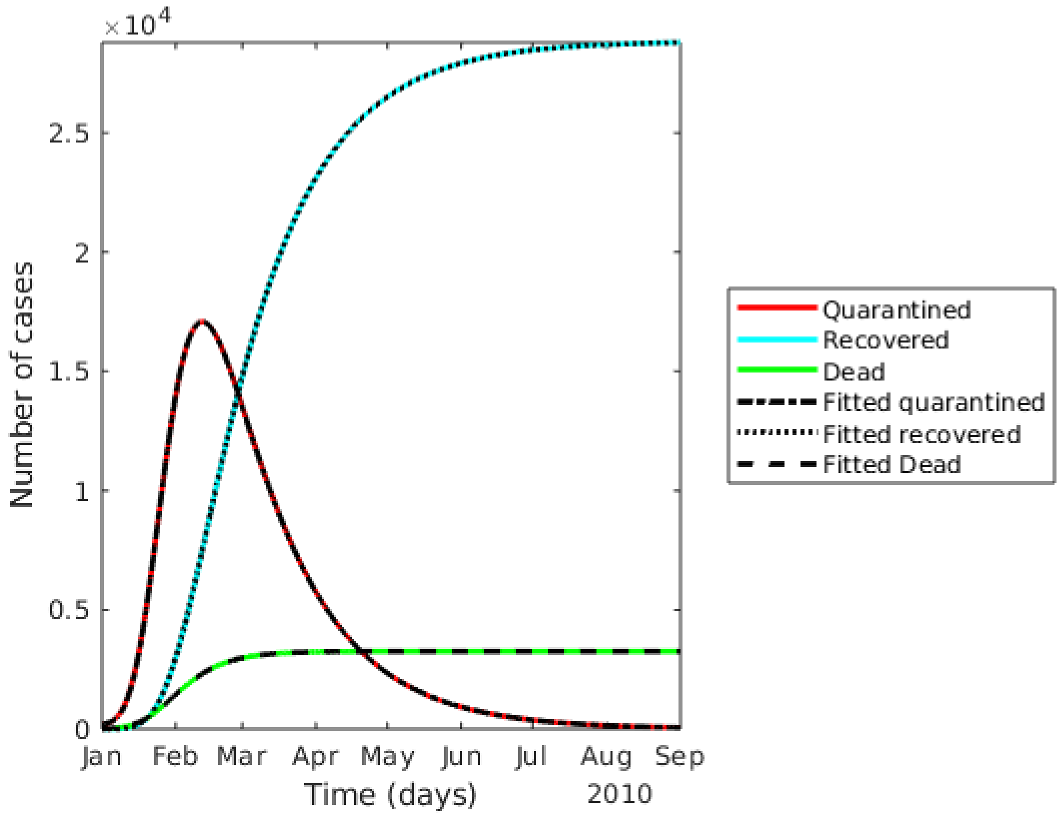
```
                                              Norm of      First-order 
 Iteration  Func-count     f(x)               step          optimality
     0           9     3.21013e+13                          4.14e+15
     1          18     4.29202e+12      0.0144713           5.7e+14
     2          27     4.12496e+11      0.0200538          7.94e+13
     3          36     2.30992e+10      0.0438648           9.2e+12
     4          45     5.52474e+09      0.0369572          1.98e+11
     5          54     5.10362e+08       0.223926          6.51e+10
     6          63     1.01745e+07      0.0543787          2.94e+10
     7          72      2.8199e+06      0.0072875          1.61e+09
     8          81      2.8199e+06      0.0251752          1.61e+09
     9          90     1.76025e+06     0.00629379          1.08e+10
    10          99      1.3007e+06      0.0138982          4.67e+10
    11         108          171652     0.00240142          8.53e+08
    12         117          103078      0.0031469           2.1e+09
    13         126         93322.7     0.00133647          3.52e+08
    14         135         93100.7     0.00135521          3.26e+07
    15         144         92985.9     0.00188907          6.28e+07
    16         153         92819.3     0.00102872          1.84e+07
    17         162         92819.3     0.00351482          1.84e+07
    18         171         92733.2    0.000878704          1.13e+07
    19         180         92615.8     0.00175741          5.09e+07
    20         189         92446.4     0.00128161          2.84e+07
    21         198           92307     0.00175741           5.2e+07
    22         207         92127.7     0.00136266          3.19e+07
    23         216         91974.4     0.00175741          5.17e+07
    24         225         91784.8     0.00144269          3.55e+07
    25         234         91616.9     0.00175741          5.15e+07
    26         243           91417     0.00152431          3.94e+07
    27         252         91246.2       0.002081          7.33e+07
    28         261         91011.5     0.00115998          2.27e+07
    29         270         90839.8     0.00175741             5e+07
    30         279         90619.7       0.001681          4.73e+07
    31         288         90404.4      0.0018959             6e+07
    32         297         90164.4     0.00153388          3.91e+07
    33         306         89954.2     0.00175741          4.96e+07
    34         315         89715.5     0.00184934          5.65e+07
    35         324         89463.7     0.00172366          4.89e+07
    36         333         89223.1     0.00202371          6.71e+07
    37         342         88948.2     0.00150875          3.72e+07
    38         351         88819.6      0.0027437          1.22e+08
    39         360         88427.2    0.000685925          5.99e+06
    40         369         88242.1     0.00137185          2.56e+07
    41         378         88102.2      0.0027437          1.19e+08
    42         387         87710.6    0.000685925          6.06e+06
    43         396         87517.7     0.00137185          2.42e+07
    44         405         87358.5      0.0027437          1.18e+08
    45         414         86969.2    0.000685925          6.11e+06
    46         423         86769.1     0.00137185          2.31e+07
    47         432         86591.8      0.0027437          1.16e+08
    48         441           86205    0.000685925          6.15e+06
    49         450         85998.2     0.00137185          2.21e+07
    50         459           85804      0.0027437          1.14e+08
    51         468         85419.8    0.000685925          6.17e+06
    52         477           85207     0.00137185          2.12e+07
```

```
    53        486        84997      0.0027437       1.13e+08
    54        495      84615.7    0.000685925       6.19e+06
    55        504      84397.5     0.00137185       2.05e+07
    56        513      84172.8      0.0027437       1.11e+08
    57        522      83794.6    0.000685925        6.2e+06
    58        531      83571.5     0.00137185       2.03e+07
    59        540      83333.4      0.0027437        1.1e+08
    60        549      82881.9     0.00114541       1.97e+07
    61        558      82642.5      0.0027437       1.11e+08
    62        567      82189.4     0.00117418       2.06e+07
    63        576      81939.8      0.0027437        1.1e+08
    64        585      81484.6     0.00120068       2.13e+07
    65        594      81225.6      0.0027437       1.09e+08
    66        603      80768.8     0.00122606        2.2e+07
    67        612      80501.1      0.0027437       1.08e+08
    68        621        80043     0.00125014       2.27e+07
    69        630      79767.5      0.0027437       1.07e+08
    70        639      79308.6     0.00127287       2.34e+07
    71        648      79026.1      0.0027437       1.06e+08
    72        657        78567     0.00129405       2.39e+07
    73        666      78278.2      0.0027437       1.05e+08
    74        675      77819.4     0.00131382       2.44e+07
    75        684      77525.4      0.0027437       1.04e+08
    76        693      77067.5     0.00133188       2.49e+07
    77        702      76768.9      0.0027437       1.03e+08
    78        711      76312.5     0.00134823       2.53e+07
    79        720      76010.4      0.0027437       1.03e+08
    80        729      75556.1     0.00136263       2.56e+07
    81        738      75251.4      0.0027437       1.02e+08
    82        747      74799.9     0.00137518       2.58e+07
    83        756      74493.5      0.0027437       1.01e+08
    84        765      74045.4     0.00138563        2.6e+07
    85        774      73738.1      0.0027437       9.97e+07
    86        783      73294.2     0.00139405       2.61e+07
    87        792      72987.1      0.0027437       9.88e+07
    88        801        72548     0.00140014        2.6e+07

Solver stopped prematurely.

lsqcurvefit stopped because it exceeded the function evaluation limit,
options.MaxFunctionEvaluations = 8.000000e+02.
```

```
[S1,E1,I1,Q1,R1,D1,P1] =...
    SEIQRDP(alpha1,beta1,gamma1,delta1,Lambda1,Kappa1,Npop,E0,I0,Q0,R0,D0,t);
```

Comparison beween fitted and generated time histories

```
figure
clf;close all;
plot(time1,Q,'r',time1,R,'c',time1,D,'g','linewidth',2);
hold on
plot(time1,Q1,'k-.',time1,R1,'k:',time1,D1,'k--','linewidth',2);
% ylim([0,1.1*Npop])
ylabel('Number of cases')
xlabel('Time (days)')
leg = {'Quarantined','Recovered','Dead','Fitted quarantined','Fitted recovered','Fitted
legend(leg{:},'location','eastoutside')
set(gcf,'color','w')
axis tight
```



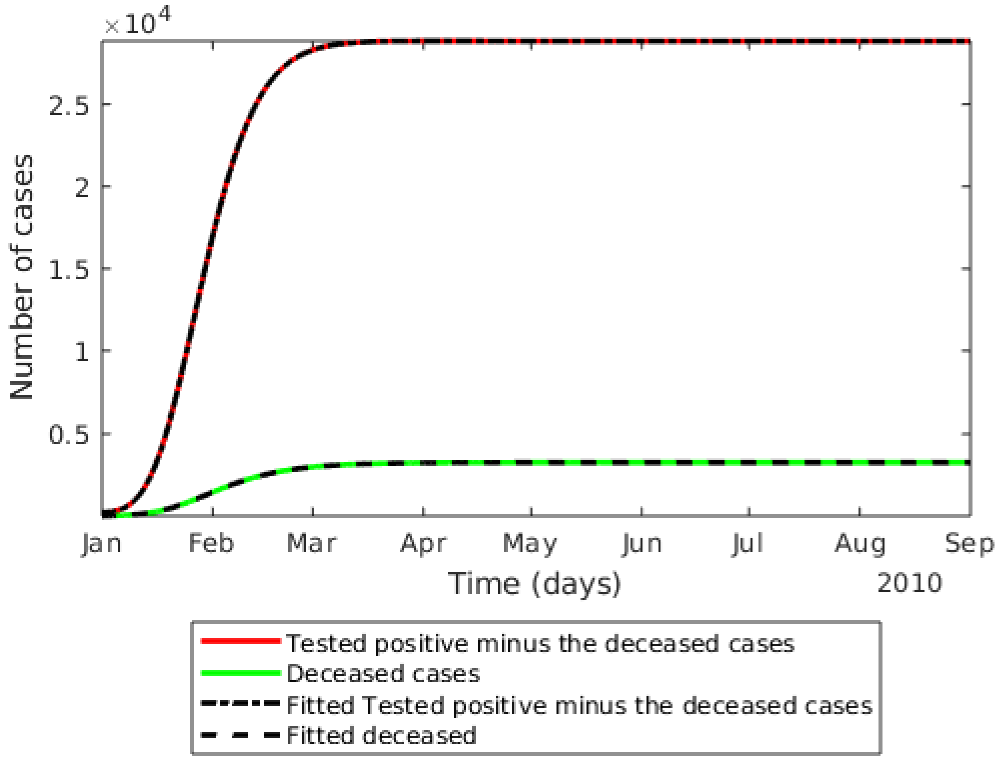
Case where the recovered (R) and quarantined (Q) data are not available separately

The number of qarantined and recovered is unknwon but Q + R is known.

```
guess = [0.05,0.9,1/4,1/10,0.03,0.03,0.02,0.06]; % initial guess

[alpha1,beta1,gamma1,delta1,Lambda1,Kappa1] = fit_SEIQRDP(Q+R,[],D,Npop,E0,I0,time1,gue
```

```
Warning:  No data available for "Recovered"
```

```

[S1,E1,I1,Q1,R1,D1,P1] =...
    SEIQRDP(alpha1,beta1,gamma1,delta1,Lambda1,Kappa1,Npop,E0,I0,Q0,R0,D0,t);

figure
clf;close all;
plot(time1,Q+R,'r',time1,D,'g','linewidth',2);
hold on
plot(time1,Q1+R1,'k-.',time1,D1,'k--','linewidth',2);
% ylim([0,1.1*Npop])
ylabel('Number of cases')
xlabel('Time (days)')
leg = {'Tested positive minus the deceased cases','Deceased cases','Fitted Tested posit
legend(leg{:},'location','southoutside')
set(gcf,'color','w')
axis tight
```


×10^4
2.5
2
1.5
1
0.5
Number of cases
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Time (days)
2010
Tested positive minus the deceased cases
Deceased cases
Fitted Tested positive minus the deceased cases
Fitted deceased