The Susceptible, Exposed, Infectious, Removed (SEIR) Model

Pierre Chaussé

September 25, 2026

Abstract

This vignette presents how to use the SEIR package

1 The SEIR Model

This package solves the SEIR model for the spread of a virus. In the example, the parameters of Wang et al. (2020) are used for the case of Wuhan, China. The model is available in Matlab on Peter Forsyth webpage (https://cs.uwaterloo.ca/ paforsyt/SEIR.html). In fact, the R package is based on the work of Peter. You are invited to read his page for more details about the model. The system of differencial equations is:

$$
\begin{aligned}
\dot{S}(t) &= -\beta(t)\frac{S(t)I(t)}{N} \\
\dot{E}(t) &= \beta(t)\frac{S(t)I(t)}{N} - \sigma E(t) \\
\dot{I}(t) &= \sigma E(t) - \gamma I(t) + c\frac{R(t)I(t)}{N} \\
\dot{R}(t) &= \gamma I(t) - c\frac{R(t)I(t)}{N}
\end{aligned}
$$

where $\beta(t) = \gamma RZero(t)$ and $RZero(t)$ is the number of individuals infected by an infectious person. It is a function of time because the transmission if function of the social distancing policy in place.

2 The Package

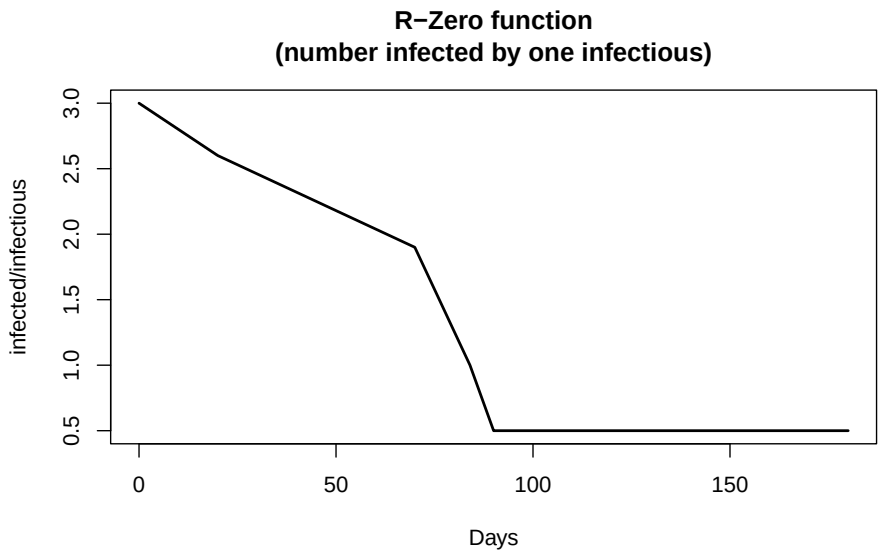
The main function is solveSEIR. It creates an object of class "seir". The main arguments are (See Peter's Matlab code for the source):

- h: The time interval used in the Runge-Kutta method.
- T: The number of days to solve.
- c: Mutation parameter (0 when no mutation is present)
- sigma: Set to 1/5.2 for the Wuhan case.
- gamma: Set to 1/18 for the Wuhan case.
- y0: The initial value $\{S(0), E(0), I(0), R(0)\}$.
- r0: The matrix of breaks for RZero(t). See below.
- type: Type of RZero(t). See below.

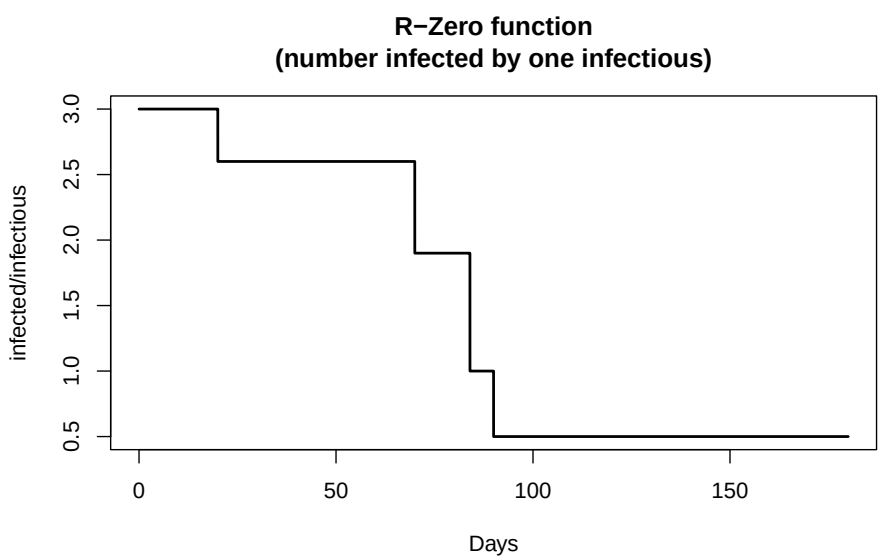
In this document, we use the default parameter used by Peter Forsyth. The important factor is $RZero(t)$, which is determined by the arguments tyee an r0. The default type is "LIN" and the default r0 is:

```r
matrix(c(0, 20, 70,
    84, 90, 3, 2.6, 1.9, 1, 0.5), ncol = 2)

##      [,1] [,2]
## [1,]    0  3.0
## [2,]   20  2.6
## [3,]   70  1.9
## [4,]   84  1.0
## [5,]   90  0.5
```

It means that the value of RZero(t) changes at t=20, 70, 84 and 90 (days). The type argument determines how it changes. By default it looks like the following:



Therefore, $RZero(t)$ is a linear interpolation between the different values. The other option is to have a constant $RZero(t)$ between periods as in the following graph.



The first $RZero(t)$ is the default option. If we run the function with all default values we obtain (the print method summarizes the result):

```
Sol <- solveSEIR()
Sol

## SEIR model for the spread of a virus
## ************************************
## Initial condition
##  Total Cases (thousands): 0.8400
##  Active Cases (thousands): 0.8400
## Condition after 180 Days
##  Total Cases (thousands): 121.4800
##  Active Cases (thousands): 4.1200
```

The object contains the following elements:

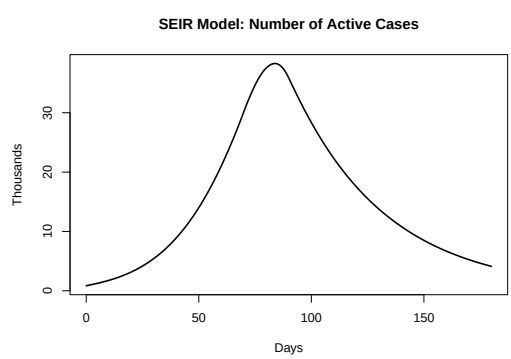
```
names(Sol)

## [1] "y"     "t"     "h"     "RZero" "Pop"
```

The element y is an $nT \times 4$ matrix, where nT is the number of grid points (the closest integer of T/h), with the ith row being the solution $\{S(t_i), E(t_i), I(t_i), R(t_i)\}$. The element t is the time grid and $RZero$ is an nT vector with the ith element being $RZero(t_i)$. The "[" method can be used to get the solution at different points in time:

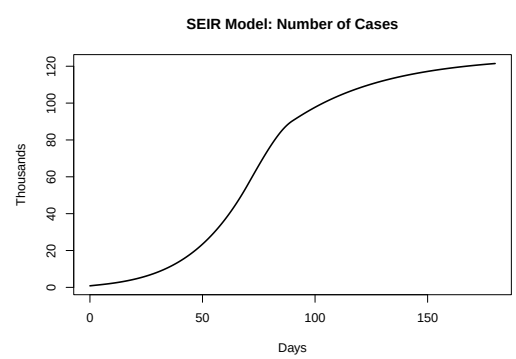
```
Sol[c(10,50,100,150)]

##            S         E         I           R
## 10  10998599  659.1349  1097.600    484.1839
## 50  10977389 4795.2626  9218.391   9437.0239
## 100 10903111 4199.9912 24144.378  69384.5890
## 150 10883650 1193.8311  7305.064 108691.5642
```

It is possible to use those elements to plot the solution, but the easiest ways is to use the plot method. The plot method gives you options to plot any of the curve:

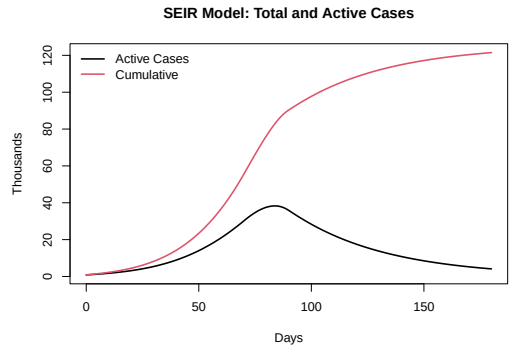
```
plot(Sol, "acase")
```



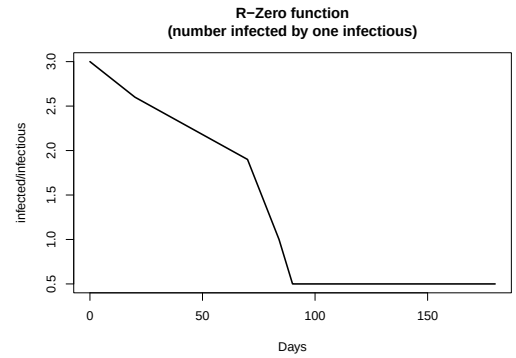

```
plot(Sol, "tcase")
```



```r
plot(Sol, "both")
```

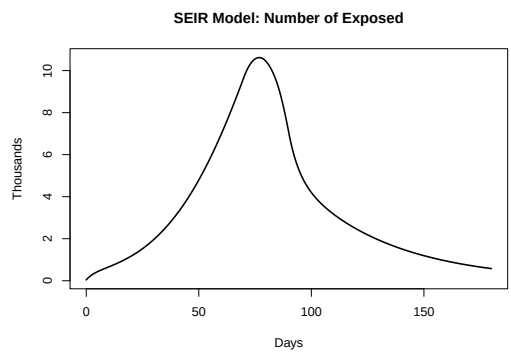



```r
plot(Sol, "rzero")
```

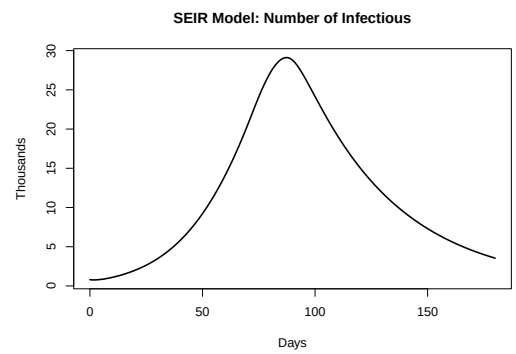


You can also plot them one by one:

```r
plot(Sol, "E")
```

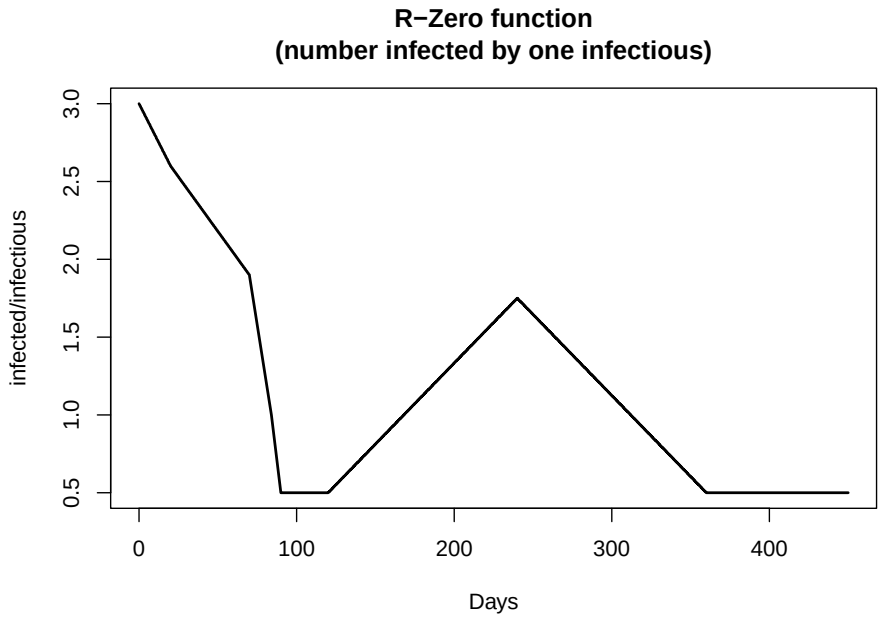



```r
plot(Sol, "I")
```



As a last example, we can reproduce the second wave case produced by Peter Forsyth on his website. Suppose RZero stays at 0.5 for 30 days and then the government start relaxing the social distancing restrictions. The following is what he proposes:

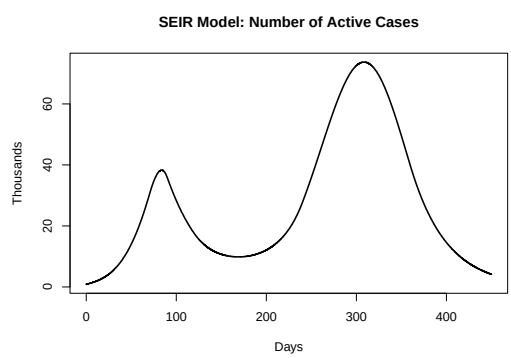
```r
zbreaks <- matrix(c(0,  20,  70,  84,  90, 120, 240, 360,
                    3, 2.6,  1.9, 1,  0.5, 0.5, 1.75, 0.5), ncol = 2)
Sol <- solveSEIR(T=450, r0=zbreaks)
plot(Sol, "rzero")
```

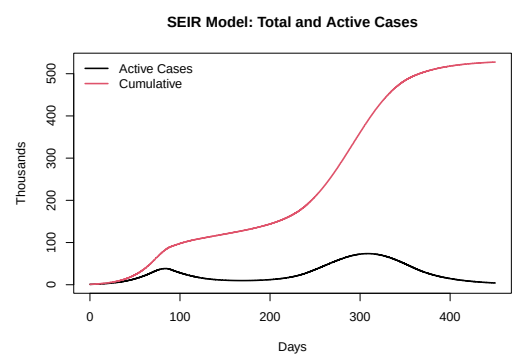


The new solution is:

```r
plot(Sol, "acase")
```

```r
plot(Sol, "both")
```





References

H. Wang, Z. Wang, Y Dong, R. Chang, C. Xu, X. Yu, S. Zhang, L. Tsamlag, M. Shang, J. Huang, Y. Wang, G. Xu, T. Shen, and X. Zhang. Phase-adjusted estimation of the number of coronavirus disease 2019 cases in wuhan, china. *Cell Discov*, 6(10), 2020. URL `https://doi.org/10.1038/s41421-020-0148-0`.