

Numerical solution of deterministic epidemiological models: Solutions

John M. Drake and Pejman Rohani

August 12, 2021

Licensed under the Creative Commons attribution-noncommercial license,
<http://creativecommons.org/licenses/by-nc/3.0/>. Please share and remix
noncommercially, mentioning its origin .

Exercise 1. Explore the dynamics of the system for different values of the β and γ parameters by simulating and plotting trajectories as time series and in phase space (i.e., I vs. S). How does the epidemic change as β becomes large? How does the epidemic change as γ becomes small? What happens if β and γ are changed together so that their ratio is constant?

First we require the `deSolve` package so that we can use the `lsoda` differential equation solver.

```
> require(deSolve)                                #deSolve library needed for this computing session
```

Here is a function that returns the derivatives of the *SIR* model.

```
> sir.model.closed <- function (t, x, params) {    #here we begin a function with three arguments
+   S <- x[1]                                     #create local variable S, the first element of x
+   I <- x[2]                                     #create local variable I
+   R <- x[3]                                     #create local variable R
+   with(                                         #we can simplify code using "with"
+     as.list(params),                            #this argument to "with" lets us use the variable names
+     {                                           #the system of rate equations
+       dS <- -beta*S*I
+       dI <- beta*S*I-gamma*I
+       dR <- gamma*I
+       dx <- c(dS,dI,dR)                        #combine results into a single vector dx
+       list(dx)                                 #return result as a list
+     }
+   )
+ }
```

Here we use `lsoda` to solve at the parameter values in the example.

```
> times <- seq(0,120,by=1)                        #function seq returns a sequence
> params <- c(beta=0.3,gamma=1/7)                #function c "c"ombines values into a vector
> xstart <- c(S=9999/10000,I=1/10000,R=0)        #initial conditions
> out <- as.data.frame(lsoda(xstart,times,sir.model.closed,params)) #result stored in dataframe
```

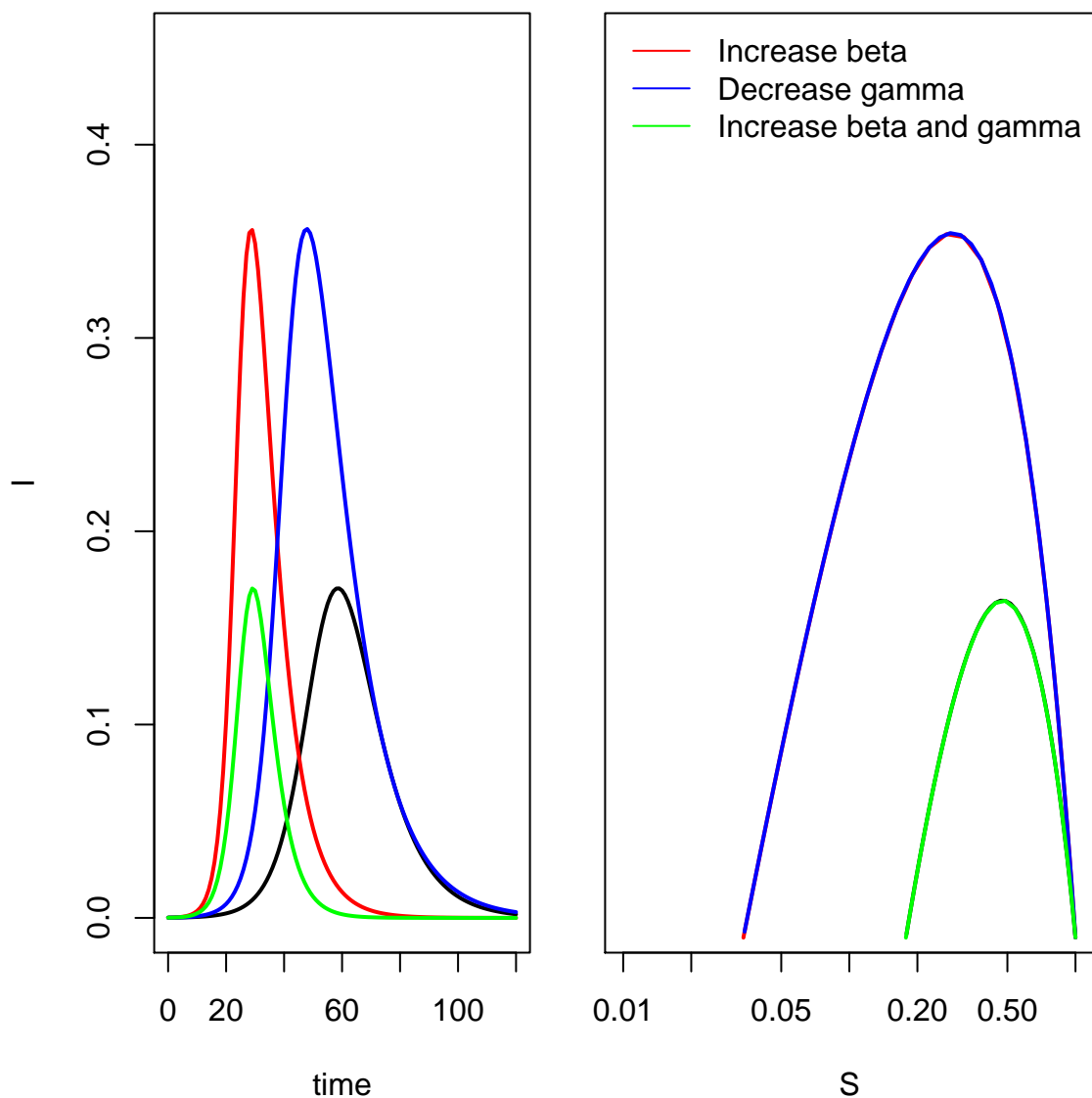
Now we solve at a larger value of β (specifically $\beta = 0.5$), a smaller value of γ ($\gamma = 0.0857$), and a pair where each has been multiplied by 2 ($\beta = 0.6$, $\gamma = 2/7$).

```
> params2 <- c(beta=0.5,gamma=1/7)
> params3 <- c(beta=0.3,gamma=0.0857)
> params4 <- c(beta=0.6,gamma=2/7)
```

Solving and plotting we see that:

1. Increasing β (red curve) causes the outbreak to happen faster (the initial growth in the increase in infected individuals is faster and the peak is higher); also the total epidemic size is larger (i.e., the area under the curve)
2. Decreasing γ (to a level that achieves the same R_0 used in the previous case; blue curve) similarly leads to a larger epidemic, however the initial takeoff is not as fast and the peak is delayed compared with increasing β . The phase portrait is the same as for the previous case, which means that the epidemic traces the same path, but does it more slowly.
3. Putting these two results together, we anticipate the result we obtain by increasing β and γ by a common factor of two (green curve): Since the R_0 is unaffected the phase portrait is not changed, but the epidemic does occur faster (since the value of β has been increased.)

```
> out2 <- as.data.frame(lsoda(xstart,times,sir.model.closed,params2))
> out3 <- as.data.frame(lsoda(xstart,times,sir.model.closed,params3))
> out4 <- as.data.frame(lsoda(xstart,times,sir.model.closed,params4))
> op <- par(fig=c(0,0.5,0,1),mar=c(4,4,1,1))
> plot(I~time,data=out,type='l', lwd=2, ylim=c(0, 0.45))
> lines(I~time,data=out2,type='l', lwd=2, col='red')
> lines(I~time,data=out3,type='l',lwd=2, col='blue')
> lines(I~time,data=out4,type='l', lwd=2, col='green')
> par(fig=c(0.5,1,0,1),mar=c(4,1,1,1),new=T)
> plot(I~S,data=out,type='l',lwd=2, log='x',yaxt='n',xlab='S', ylim=c(0.01, 0.45), xlim=c(0.01, 1))
> lines(I~S,data=out2,type='l', lwd=2, col='red')
> lines(I~S,data=out3,type='l', lwd=2, col='blue')
> lines(I~S,data=out4,type='l', lwd=2, col='green')
> legend('topleft', col=c('red','blue','green'), legend=c('Increase beta', 'Decrease gamma', 'Increase I'))
> par(op)
```



Exercise 2. Explore the dynamics of the system for one set of β and γ at different initial conditions. What happens if there is pre-existing immunity in the population?

Here we return to our initial set of parameter values ($\beta = 0.3$, $\gamma = 1/7$).

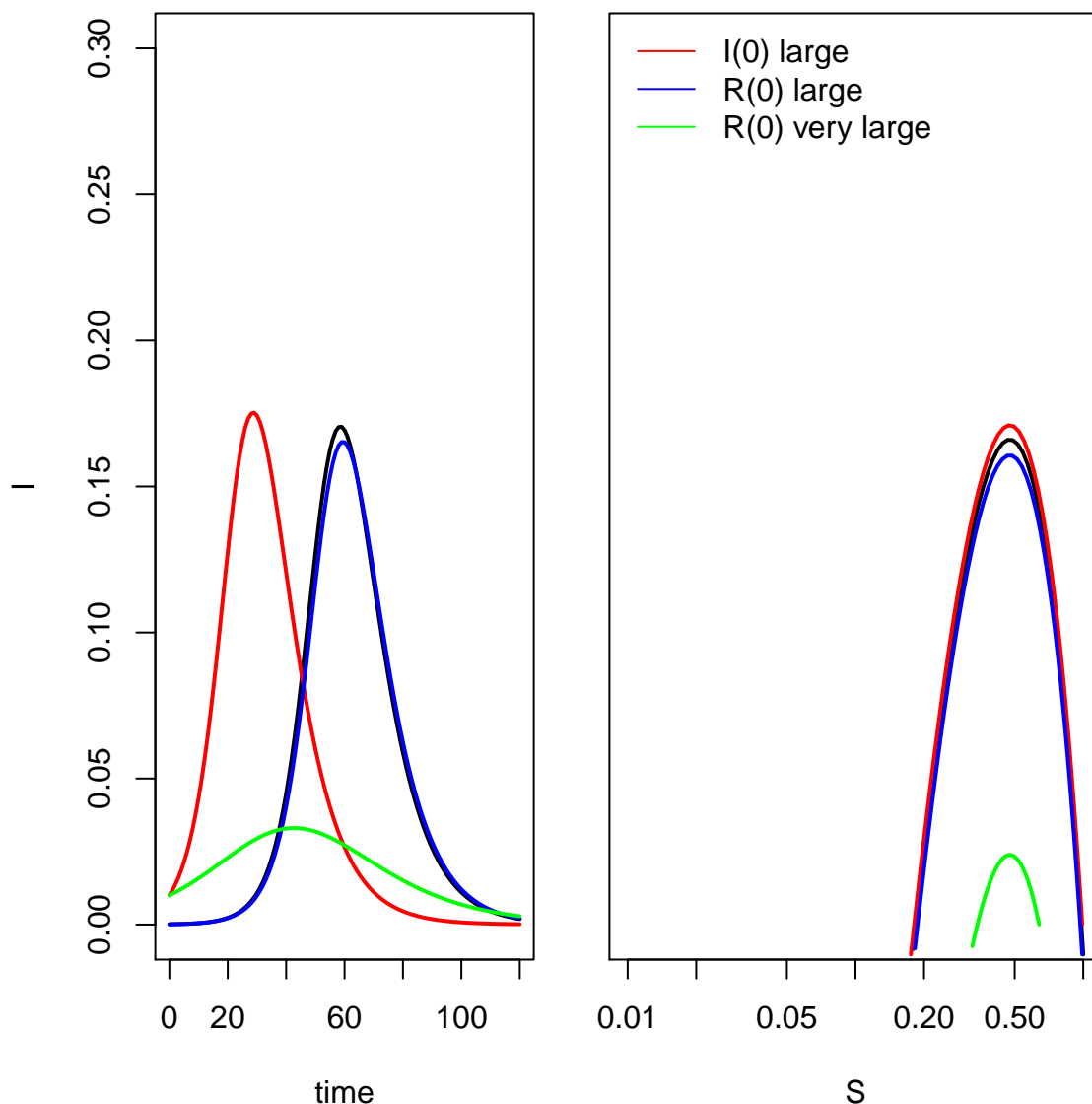
What we change are the initial conditions. We will consider a case with a hundredfold greater number of initial infectives, but no immunity in the population ($S = 9900/10000$, $I = 100/10000$, $R = 0$), a case with the original number of initial infectives and some immune individuals ($S = 9899/10000$, $I = 1/10000$, $R = 100$), and a case with a large number of initial infectives and a large number of immune individuals ($S = 6400/10000$, $I = 100/10000$, $R = 3500$).

```
> xstart5 <- c(S=9900/10000,I=100/10000,R=0)
> xstart6 <- c(S=9899/10000,I=1/10000,R=100)
```

```

> xstart7 <- c(S=6400/10000,I=100/10000,R=3500)
> out5 <- as.data.frame(lsoda(xstart5,times,sir.model.closed,params))
> out6 <- as.data.frame(lsoda(xstart6,times,sir.model.closed,params))
> out7 <- as.data.frame(lsoda(xstart7,times,sir.model.closed,params))
> op <- par(fig=c(0,0.5,0,1),mar=c(4,4,1,1))
> plot(I~time,data=out,type='l', lwd=2, ylim=c(0, 0.3))
> lines(I~time,data=out5,type='l', lwd=2, col='red')
> lines(I~time,data=out6,type='l', lwd=2, col='blue')
> lines(I~time,data=out7,type='l', lwd=2, col='green')
> par(fig=c(0.5,1,0,1),mar=c(4,1,1,1),new=T)
> plot(I~S,data=out,type='l',lwd=2, log='x',yaxt='n',xlab='S', ylim=c(0.01, 0.3), xlim=c(0.01, 1))
> lines(I~S,data=out5,type='l', lwd=2, col='red')
> lines(I~S,data=out6,type='l', lwd=2, col='blue')
> lines(I~S,data=out7,type='l', lwd=2, col='green')
> legend('topleft', col=c('red','blue','green'), legend=c('I(0) large', 'R(0) large', 'R(0) very large'))
> par(op)

```



What we see is that increasing the initial number infected increases the takeoff rate (red curve compared with black curve), that increasing the number immune by a similar amount has very low effect (blue curve compared with the black curve), and increasing the number immune by a lot significantly curtail the outbreak, i.e., it is smaller and slower than in a wholly susceptible populations (green curve compared with the black curve). In this case, we have started with a fraction immune (35%) about half of that compared with what would be required to achieve herd immunity ($1 - 1/R_0 = 1 - \gamma/\beta \approx 0.524$).

Exercise 3. Modify the codes given to study the dynamics of a demographically open *SIR* model.

To study a demographically open population we add a per capita birth rate (μ). We need to modify our derivatives function to include this rate in each equation and to modify our parameter vector to reflect the various values of μ that we wish to study.

First we write a new function for the derivatives.

```

> sir.model.open <- function (t, x, params) { #here we begin a function with three arguments
+   S <- x[1] #create local variable S, the first element of x
+   I <- x[2] #create local variable I
+   R <- x[3] #create local variable R
+   with( #we can simplify code using "with"
+     as.list(params), #this argument to "with" lets us use the variable names
+     { #the system of rate equations
+       dS <- mu*(S+I+R) -beta*S*I - mu*S
+       dI <- beta*S*I-gamma*I - mu*I
+       dR <- gamma*I - mu*R
+       dx <- c(dS,dI,dR) #combine results into a single vector dx
+       list(dx) #return result as a list
+     }
+   )
+ }

```

Now we modify the vector of input parameters. Since $\gamma = 0.7$, we can imagine that we are modeling an acute infection where parameters have units of days, i.e., an infectious period of 7 days. We will study three levels of μ : $\mu = 1/365$, $\mu = 1/3650$, $\mu = 1/36500$. These would correspond to life spans that for vertebrates are (1 year), intermediate (10 years), or long-lived (100 years).

```

> params8 <- c(beta=0.3,gamma=1/7, mu=1/52)
> params9 <- c(beta=0.3,gamma=1/7, mu=1/520)
> params10 <- c(beta=0.3,gamma=1/7, mu=1/5200)

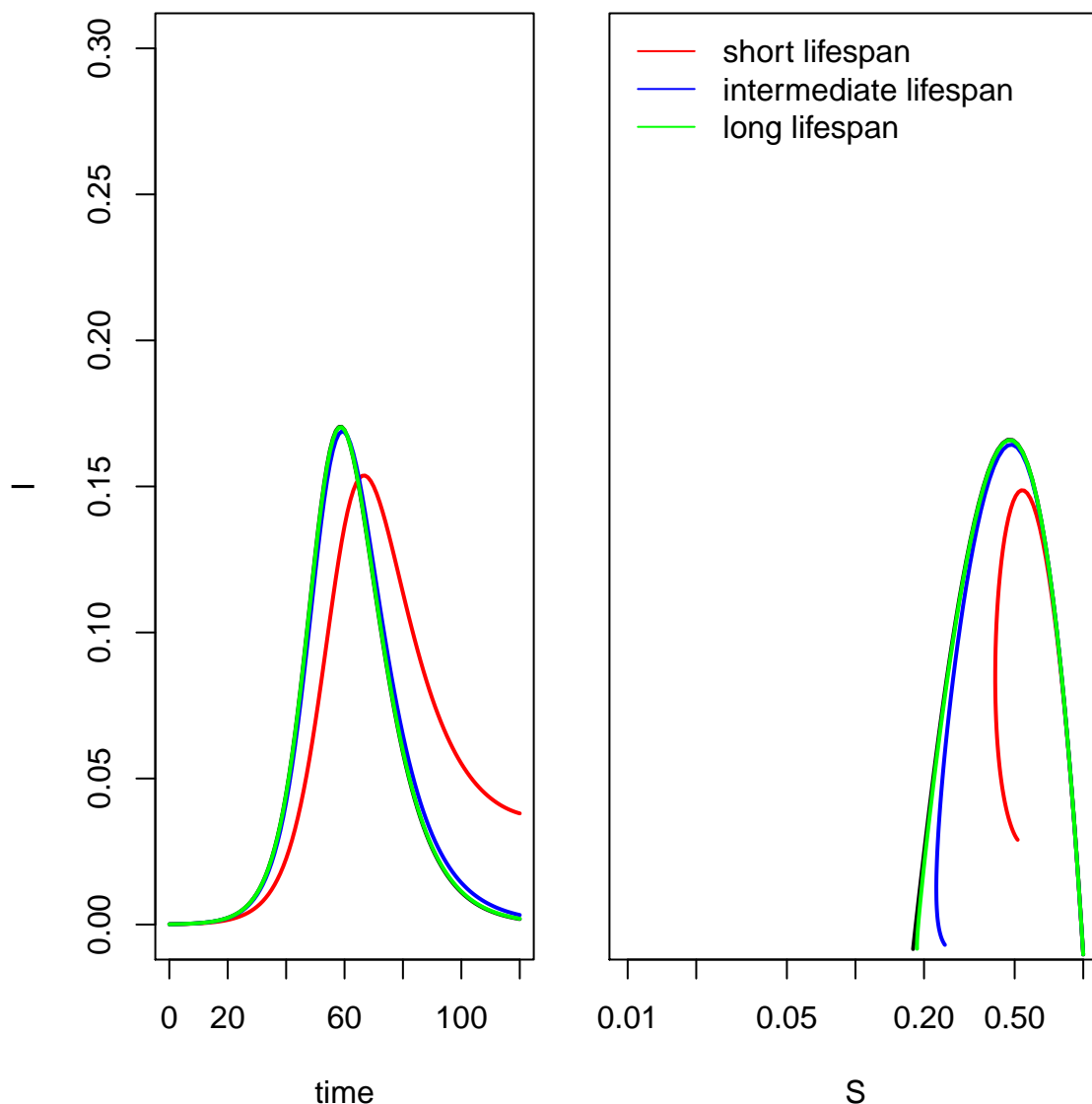
```

Finally, we solve and plot.

```

> out8 <- as.data.frame(lsoda(xstart,times,sir.model.open,params8))
> out9 <- as.data.frame(lsoda(xstart,times,sir.model.open,params9))
> out10 <- as.data.frame(lsoda(xstart,times,sir.model.open,params10))
> op <- par(fig=c(0,0.5,0,1),mar=c(4,4,1,1))
> plot(I~time,data=out,type='l', lwd=2, ylim=c(0, 0.3))
> lines(I~time,data=out8,type='l', lwd=2, col='red')
> lines(I~time,data=out9,type='l', lwd=2, col='blue')
> lines(I~time,data=out10,type='l', lwd=2, col='green')
> par(fig=c(0.5,1,0,1),mar=c(4,1,1,1),new=T)
> plot(I~S,data=out,type='l',lwd=2, log='x',yaxt='n',xlab='S', ylim=c(0.01, 0.3), xlim=c(0.01, 1))
> lines(I~S,data=out8,type='l', lwd=2, col='red')
> lines(I~S,data=out9,type='l', lwd=2, col='blue')
> lines(I~S,data=out10,type='l', lwd=2, col='green')
> legend('topleft', col=c('red','blue','green'), legend=c('short lifespan', 'intermediate lifespan', 'long lifespan'))
> par(op)

```



We see that the addition of demography, which is effectively a continuous supply of susceptible individuals, leads to an *endemic equilibrium* in which the pathogen persists in the population at a low level. The level of the endemic equilibrium declines with host life span in a population of stable size. The intuitive explanation for this phenomenon is that those populations with shorter life span have higher turnover and therefore a greater flow of susceptible individuals. Particularly, the open *SIR* model recovers the closed *SIR* model as a limit case as $\mu \rightarrow \infty$. Thus, the outbreaks in populations with intermediate and long life span look very much like the closed *SIR* epidemic.

***Exercise 4.** Modify the codes given to study the dynamics of an *SEIR* model.

Here we modify the derivatives function to include an additional equation and an additional state variable. Additionally, we will have to pass a new parameter (the inverse of latent period) and an additional initial state variable (the number of individuals in latent class at start of the epidemic). We study the *SEIR* epidemic at latent periods of 0.0001 days (nearly equivalent to the *SIR* epidemic), 3.5 days (half of the

infectious period), 7 days (equal to the infectious period), and 14 days (double the infectious period).

First, we write our new function.

```
> seir.model.closed <- function (t, x, params) {
+   S <- x[1]                                #create local variable S
+   E <- x[2]                                #create local variable S
+   I <- x[3]                                #create local variable I
+   R <- x[4]                                #create local variable R
+   with(                                     #we can simplify code using "with"
+     as.list(params),                       #this argument to "with" lets us use the variable names
+     {                                       #the system of rate equations
+       dS <- -beta*S*I
+       dE <- beta*S*I - sigma*E
+       dI <- sigma*E - gamma*I
+       dR <- gamma*I
+       dx <- c(dS,dE, dI,dR)                #combine results into a single vector dx
+       list(dx)                             #return result as a list
+     }
+   )
+ }
```

Now we set our new parameter vectors.

```
> params11 <- c(beta=0.3, sigma=1/0.0001, gamma=1/7)
> params12 <- c(beta=0.3, sigma=1/3.5, gamma=1/7)
> params13 <- c(beta=0.3, sigma=1/7, gamma=1/7)
> params14 <- c(beta=0.3, sigma=1/14, gamma=1/7)
```

We set a new initial condition.

```
> xstart <- c(S=9999/10000, E=0, I=1/10000,R=0)    #initial conditions
```

As we will see, these epidemics last longer. Therefore, we will also extend the interval over which we solve.

```
> times <- seq(0,240,by=1)                        #function seq returns a sequence
```

Finally, we solve and plot.

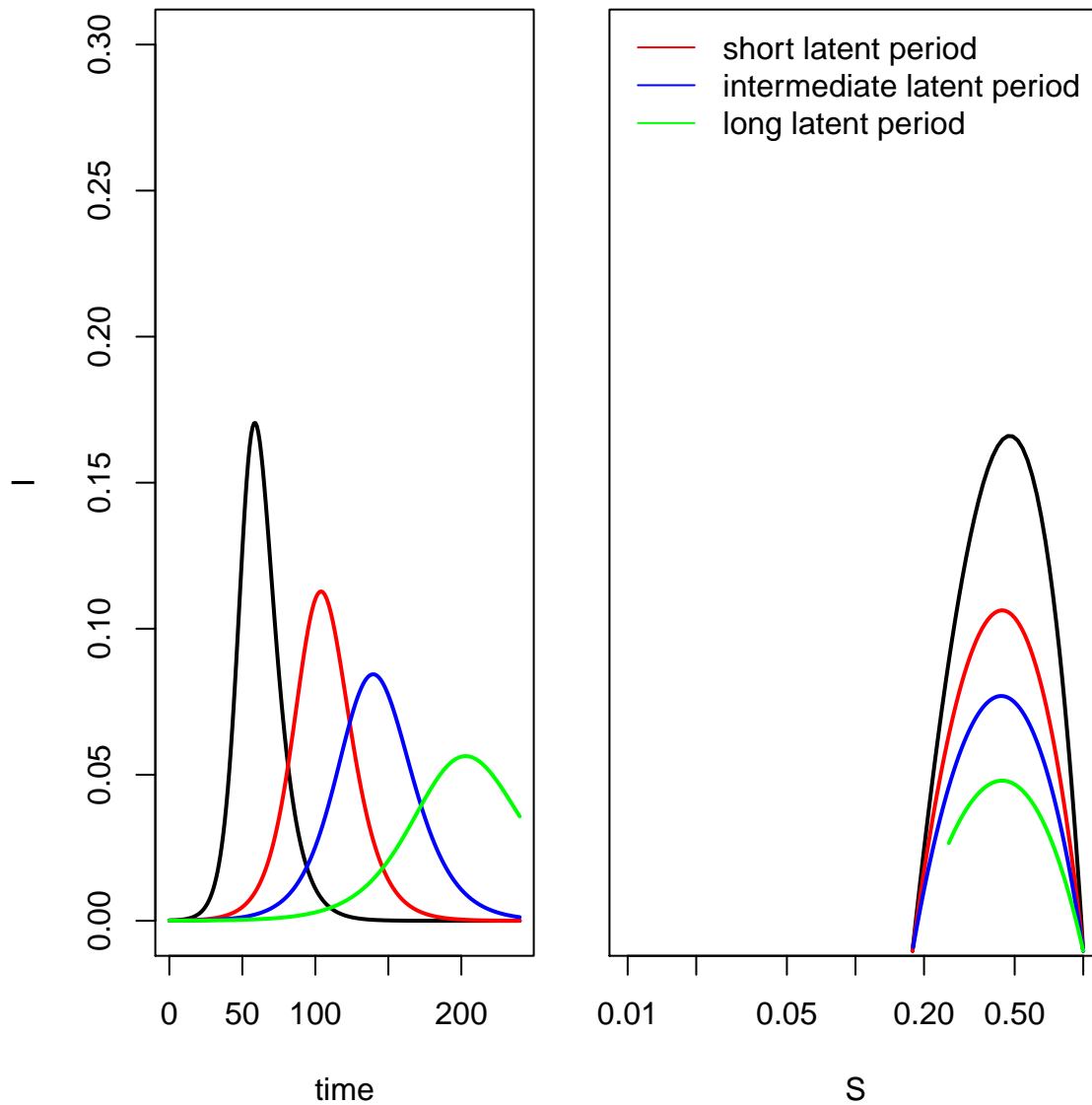
```
> out11 <- as.data.frame(lsoda(xstart,times,seir.model.closed,params11))
> out12 <- as.data.frame(lsoda(xstart,times,seir.model.closed,params12))
> out13 <- as.data.frame(lsoda(xstart,times,seir.model.closed,params13))
> out14 <- as.data.frame(lsoda(xstart,times,seir.model.closed,params14))
> op <- par(fig=c(0,0.5,0,1),mar=c(4,4,1,1))
> plot(I~time,data=out11,type='l', lwd=2, ylim=c(0, 0.3))
> lines(I~time,data=out12,type='l', lwd=2, col='red')
> lines(I~time,data=out13,type='l', lwd=2, col='blue')
> lines(I~time,data=out14,type='l', lwd=2, col='green')
```



```

> par(fig=c(0.5,1,0,1),mar=c(4,1,1,1),new=T)
> plot(I~S,data=out11,type='l',lwd=2, log='x',yaxt='n',xlab='S', ylim=c(0.01, 0.3), xlim=c(0.01, 1))
> lines(I~S,data=out12,type='l', lwd=2, col='red')
> lines(I~S,data=out13,type='l', lwd=2, col='blue')
> lines(I~S,data=out14,type='l', lwd=2, col='green')
> legend('topleft', col=c('red','blue','green'), legend=c('short latent period', 'intermediate latent p
> par(op)

```



We see that the addition of a latent period delays the epidemic takeoff and “stretches” the outbreak over a longer period of time.