Seasonally forced epidemics
Biennial outbreaks of Measles in England and Wales *

John M. Drake & Pejman Rohani

1 Introduction

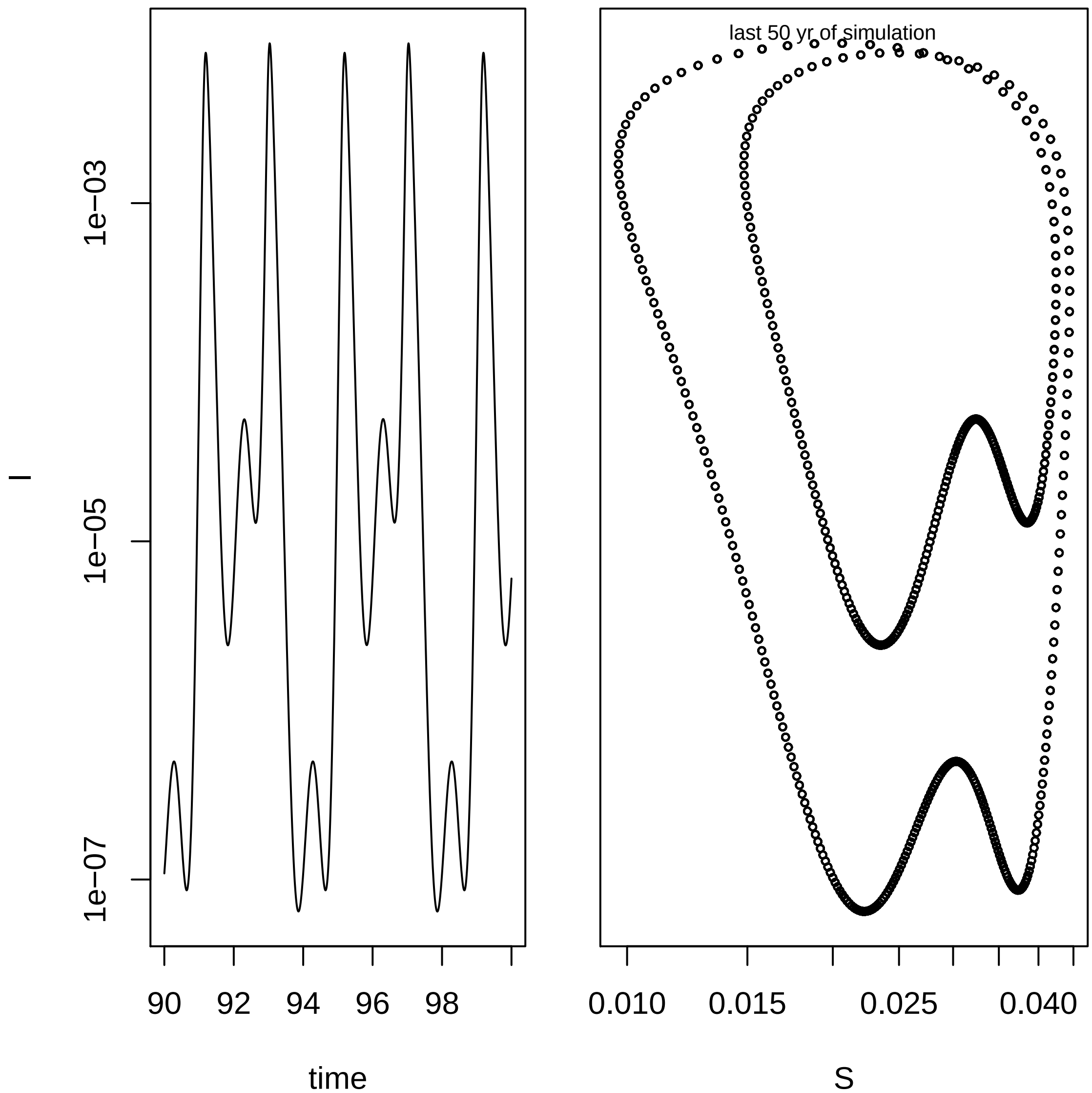
As we have now seen on several occasions, the simple SIR model predicts damped oscillations towards an equilibrium (or pathogen extinction if R_0 is too small). This is at odds with the recurrent outbreaks seen in many real pathogens, for instance dynamics of measles in England and Wales prior to vaccination as studied in the last exercise. In general, sustained oscillations require some additional seasonal "driver" in the model. An important driver in human infections is changes in contact rates due to aggregation of children the during school term. We can analyze the consequences of this by assuming sinusoidal forcing on β such as to $\beta(t) = \beta_0 \,(1 + \beta_1 \cos(2\,\pi\,t))$. Translating this into R:

```
> seasonal.sir.model <- function (t, x, params) {  #function to return time-dependent rates
+   with(
+        as.list(c(x,params)),
+        {
+          beta <- beta0*(1+beta1*cos(2*pi*t))      #first determine time-dependent beta
+          dS <- mu*(1-S)-beta*S*I                  #the system of rate equations
+          dI <- beta*S*I-(mu+gamma)*I
+          dR <- gamma*I-mu*R
+          dx <- c(dS,dI,dR)                        #store result
+          list(dx)                                 #and return as a list
+        }
+        )
+ }
```

Here we simulate with the same mean contact rate, β_0, as before, but now with a fairly strong amplitude of seasonality, β_1.

```
> require(deSolve)
> times <- seq(0,100,by=1/120)                      #times at which to obtain solution
> params <- c(mu=1/50,beta0=1000,beta1=0.4,gamma=365/13)   #parameters
> xstart <- c(S=0.06,I=0.001,R=0.939)               #initial conditions
> out <- as.data.frame(lsoda(xstart,times,seasonal.sir.model,params,rtol=1e-12,hmax=1/120)) #solve
> op <- par(fig=c(0,0.5,0,1),mar=c(4,4,1,1))        #set graphical parameters
> plot(I~time,data=out,type='l',log='y',subset=time>=90) #plot
> par(fig=c(0.5,1,0,1),mar=c(4,1,1,1),new=T)        #reset graphical parameters
```

```
> plot(I~S,data=out,type='p',log='xy',subset=time>=50,yaxt='n',xlab='S',cex=0.5) #plot
> text(0.02,1e-02,cex=0.7, "last 50 yr of simulation")  #annotate graph
> par(op)                                          #reset graphical parameters
```



Exercise 1. Explore the effects of changing amplitude of seasonality, β_1 on the dynamics of this model. Be careful to distinguish between transient and asymptotic dynamics.

***Exercise 2.** Suppose it is reproduction, not transmission, that is seasonally dependent. Modify the codes given to study the dynamics of an SIR model with birth pulses.

***Exercise 3.** Adapt the stochastic SIR model to include seasonal forcing.

Seasonality in the age-structured model

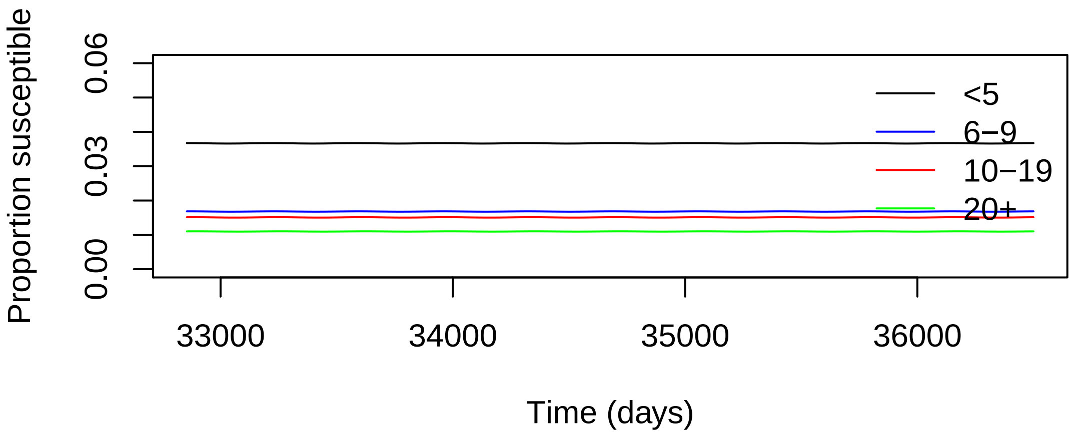
We conclude this exercise by trying to fully understand the causes of the sustained biennial measles cycle from England and Wales. This will require bringing together three topics we have covered in this course and which are now in our collective repertoire: solution of ODEs, age-structured heterogeneities, and seasonal forcing. As we did for the standard SIR model, in this section we add seasonality in the form of a sin wave. Since the biological cause of seasonlity is increased transmission among school children when school is in session, our approach will be to make the $\beta_{2,2}$ term a time-varying parameter. All other parameters remain the same.

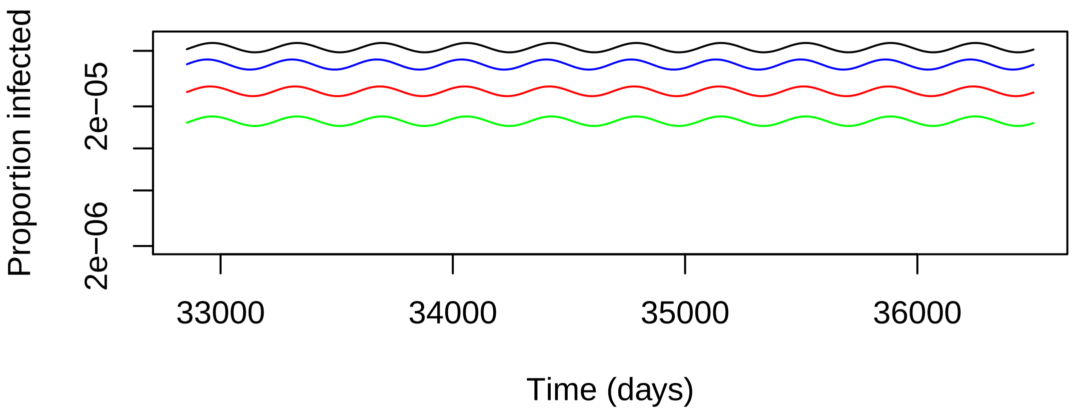
```
> seasonal.age.model<-function(t,x,parms){
+  S<-x[1:4]
+  E<-x[5:8]
+  I<-x[9:12]
+  dx<-vector(length=12)
+  beta<-parms$beta
+  beta[2,2]<-parms$beta[2,2]*(1+0.04*cos(2*pi*t/365))
+  #at all times update transmission matrix with cosine function
+  day<-floor(t)  #let day e integerized time
+  for(a in 1:4){
+    tmp <- (beta[a,]%*%I)*S[a]
+    dx[a] <- parms$nu[a]*55/75 - tmp - parms$mu[a]*S[a]
+    dx[a+4] <- tmp - parms$sigma*E[a] - parms$mu[a]*E[a]
+    dx[a+8] <- parms$sigma*E[a] - parms$gamma*I[a] - parms$mu[a]*I[a]
+  }
+ return(list(dx))
+ }
```

Finally, we solve the equations one year at a time, moving age classes up between years, and plot.

```
>   y0<-c(0.05, 0.01, 0.01, 0.008, 0.0001, 0.0001, 0.0001, 0.0001, 0.0001, 0.0001, 0.0001, 0.0001)
> parms<-list(beta=matrix(c(2.089, 2.089, 2.086, 2.037, 2.089, 9.336, 2.086, 2.037, 2.086, 2.086,
+             2.086, 2.037, 2.037, 2.037, 2.037,2.037),nrow=4,byrow=TRUE),
+             sigma=1/8, gamma=1/5, nu=c(1/(55*365),0,0,0),  mu=c(1/(55*365),0,0,0))
> n=c(6,4,10,55)/75   #number of years in each age class
> maxTime <- 100*365  #number of days in 100 years
> T0=0
> S=c()
> E=c()
> I=c()
> T=c()
> while(T0<maxTime){
+   y=lsoda(y0,seq(T0, T0+365,by=5),seasonal.age.model,parms)
+   T=rbind(T, y[2,1])
+   S=rbind(S, y[2,2:5])
+   E=rbind(E, y[2,6:9])
+   I=rbind(I, y[2,10:13])
+   y0[1]=tail(y,1)[2]-tail(y,1)[2]/6
+   y0[2]=tail(y,1)[3]+tail(y,1)[2]/6 - tail(y,1)[3]/4
+   y0[3]=tail(y,1)[4]+tail(y,1)[3]/4 - tail(y,1)[4]/10
+   y0[4]=tail(y,1)[5]+tail(y,1)[4]/10
+   y0[5]=tail(y,1)[6]-tail(y,1)[6]/6
+   y0[6]=tail(y,1)[7]+tail(y,1)[6]/6 - tail(y,1)[7]/4
+   y0[7]=tail(y,1)[8]+tail(y,1)[7]/4 - tail(y,1)[8]/10
+   y0[8]=tail(y,1)[9]+tail(y,1)[8]/10
+   y0[9]=tail(y,1)[10]-tail(y,1)[10]/6
```

```
+   y0[10]=tail(y,1)[11]+tail(y,1)[10]/6 - tail(y,1)[11]/4;
+   y0[11]=tail(y,1)[12]+tail(y,1)[11]/4 - tail(y,1)[12]/10;
+   y0[12]=tail(y,1)[13]+tail(y,1)[12]/10
+   T0=tail(T,1)
+ }
> par(mfrow=c(2,1))
> plot(tail(T,730),tail(S[,1],730),type='l',ylim=c(0,0.06),
+      xlab='Time (days)',ylab='Proportion susceptible')
> lines(tail(T,730),tail(S[,2],730),col='blue')
> lines(tail(T,730),tail(S[,3],730),col='red')
> lines(tail(T,730),tail(S[,4],730),col='green')
> legend(x='topright',legend=c('<5','6-9','10-19','20+'),
+        col=c('black','blue','red','green'),lty=1,bty='n')
> plot(tail(T,730),tail(I[,1],730),type='l',log='y',ylim=c(2e-6,6e-5),
+      xlab='Time (days)',ylab='Proportion infected')
> lines(tail(T,730),tail(I[,2],730),col='blue')
> lines(tail(T,730),tail(I[,3],730),col='red')
> lines(tail(T,730),tail(I[,4],730),col='green')
>
```





Cycles!

***Exercise 4.** If the summer break in school is the only holiday long enough to affect transmission, then the seasonally forced model developed above might be expected to re-capture the biennial dynamics observed at the beginning of this session. Instead, the model dynamics appear to be annual. One possibility is that the shorter holidays (Christmas, Easter, and autumn half term) also play a role. Modify the seasonal model to account for these holidays by changing the within-class transmission term from the constant model so that it is 9.336+4.571 when school is in term and 9.336-4.571 when school is not in term. The complete set of holidays is days 1-6, 100-115, 200-251, and 300-307.