Practical 3: Compartmental Models to Equations

Sam Abbott

# Learning Objectives

1. Know how to translate a simple model flow diagram into equations.
2. Know how to use a simple model scaffold to develop a more complex model.
3. Understand how to implement your own model structure.

# Outline for Session

1. Set up (5 minutes)
2. Practise simulating a fully implemented SEIR model (10 minutes).
3. Add high and low risk latency to the SEIR model (10 minutes).
4. Translate a more realistic SHLIR model flow diagram to equations (10 minutes).
5. Implement your own model into R (20 minutes).
6. Session wrap up (5 minutes)

# Exercise

## 1. A Simple SEIR Model of Tuberculosis (TB)

As a first exercise we are going to run the simple SEIR model, as seen in practical 2, in R. If you have not installed the course package do this now with the following R code (speak to an instructor if you are having issues with this step).

install.packages("devtools")  
devtools::install\_github("bristolmathmodellers/biddmodellingcourse")

Now load the course package,

library(biddmodellingcourse)

For more help getting started see the course README (<https://bristolmathmodellers.github.io/biddmodellingcourse/>) or ask an instructor.

### Populations and Initialisation

We first set up the initial populations for the Susceptible (), Latent (), Infected (), and Recovered () compartments. We have initialised the model as an early stage epidemic with a single case of TB.

inits = c(  
 S = 999,  
 E = 0,  
 I = 1,  
 R = 0  
)

### Parameters

We then specify the model parameters (with the units being years-1), varying these parameters will impact the model dynamics.

parameters <- c(  
 beta = 3, # Rate of transmission  
 gamma = 1/5, # Rate of progression to active symptoms   
 tau = 1/2, # Rate of recovery  
 mu = 0 # Rate of natural mortality  
)

### Equations

Finally we specify the model equations for each population compartment. This model incorporates simple demographic processes with a constant natural death rate from all compartments which is equal to the birth rate into the susceptible compartment.

SEIR\_demo\_ode <- function(t, x, params) {  
  
 ## Specify model compartments  
 S <- x[1]  
 E <- x[2]  
 I <- x[3]  
 R <- x[4]  
  
 with(as.list(params),{  
  
 ## Specify total population  
 N = S + E + I + R  
  
 ## Derivative Expressions  
 dS = -beta \* S \* I / N - mu \* S + mu \* N  
 dE = beta \* S \* I / N - gamma \* E - mu \* E  
 dI = gamma \* E - tau \* I - mu \* I  
 dR = tau \* I - mu \* R  
  
 ## output  
 derivatives <- c(dS, dE, dI, dR)  
  
 list(derivatives)  
 })  
}

### Simulate and Summarise

To simulate the model we specify the starting year (begin\_time) and final year (end\_time) and define a sequence over all intervening years. The model is then solved using deSolve::lsoda which is used within a simple wrapper function (see ?solve\_ode for details). The resulting table summarises the simulation results for the first 5 years.

begin\_time <- 0  
end\_time <- 200  
times <- seq(begin\_time, end\_time, 1)  
  
## see ?solve\_ode for details  
SEIR\_sim <- biddmodellingcourse::solve\_ode(model = SEIR\_demo\_ode,   
 inits = inits,   
 params = parameters,  
 times = times,   
 as.data.frame = TRUE)  
SEIR\_sim %>%   
 head %>%   
 pretty\_table(caption = "First 5 years of model simulation",  
 label = "tab-1")

We then summarise the epidemic peak (epi\_peak) and epidemic duration (epi\_dur), along with population sizes at the end of the time period simulated.

biddmodellingcourse::summarise\_model(SEIR\_sim) %>%   
 pretty\_table(caption = "SEIR model summary statistics; The final size of each population at the end of the simulation, along with the time the epidemic peak was reached, the number of infected at the epidemic peak and the duration of the epidemic")

Finally we plot the population in each model compartment over time.

## For an interactive graph change interactive to TRUE  
## Interactivity allows plot zooming and gives a tool tip providing the population sizr at any point.  
biddmodellingcourse::plot\_model(SEIR\_sim, interactive = FALSE)

### Explore

As we saw in practical 2, model dynamics are parameter dependent. Look back at the questions from practical 2 and check that you can implement the changes in the code above to answer them.

# 2. Add High and Low Risk Compartments

Now we are going to implement the SHLIR model from practical 2 and try to reproduce some of the behaviour observed using the interactive interface. See practical for details on this model.

### Populations and Initialisation

As in the previous model the first step is to define the model populations. There are now two new compartments, high risk latents (H), and low risk latents (L). These replace the original latent population (E) used in the SEIR model.

SHLIR\_inits = c(  
 S = 999,  
 H = 0,  
 L = 0,  
 I = 1,  
 R = 0  
)

### Parameters

We add two additional parameters for the rate of progression from high to low risk latents (nu) and the rate of progression from low risk latent to active disease (gamma\_L).

SHLIR\_parameters <- c(  
 beta = 3, # Rate of transmission  
 gamma\_H = 1/5, # Rate of progression to active symptoms from high risk latent  
 nu = 1/2, #Rate of progression from high to low risk latent  
 gamma\_L = 1/100, # Rate of progression to active symptoms for low risk latent  
 tau = 1/2, # Rate of recovery  
 mu = 1/81 # Rate of natural mortality  
)

### Equations

The code below is a starting point, fill in the missing model terms using the model flow diagram (Figure 1) and the code for the previous SEIR model as reference points.

SHLIR\_demo\_ode <- function(t, x, params) {  
  
 ## Specify model compartments  
 S <- x[1]  
 H <- x[2]  
 L <- x[3]  
 I <- x[4]  
 R <- x[5]  
  
 with(as.list(params),{  
  
 ## Specify total population  
 N = S + H + L + I + R  
  
 ## Derivative Expressions  
 dS = - beta \* S \* I / N - mu \* S + mu \* N  
 ## These are the new equations - fill in the remaining terms  
 dH =   
 dL =   
 ## Hint terms are missing from this equation as well  
 dI = gamma\_H \* H + gamma\_L \* L - tau \* I  
 dR =   
  
 ## output  
 derivatives <- c(dS, dH, dL, dI, dR)  
  
 list(derivatives)  
 })  
}

### Simulate and Summarise

Simulation is the same as for the previous model. Does the simulation of your improved model make sense? Evaluate the summary tables and plot of model populations over time.

begin\_time <- 0  
end\_time <- 200  
SHILR\_times <- seq(begin\_time, end\_time, 1)  
  
## see ?solve\_ode for details  
SHLIR\_sim <- biddmodellingcourse::solve\_ode(model = SHLIR\_demo\_ode,   
 inits = SHLIR\_inits,   
 params = SHLIR\_parameters,  
 times = SHILR\_times,   
 as.data.frame = TRUE)  
SHLIR\_sim %>%   
 head %>%   
 pretty\_table(caption = "First 5 years of SHLIR model simulation",  
 label = "tab-1")

biddmodellingcourse::summarise\_model(SHLIR\_sim) %>%   
 pretty\_table(caption = "SHLIR model summary statistics; The final size of each population at the end of the simulation, along with the time the epidemic peak was reached, the number of infected at the epidemic peak and the duration of the epidemic")

## For an interactive graph change interactive to TRUE  
biddmodellingcourse::plot\_model(SHLIR\_sim, interactive = FALSE)

### Explore

1. Test your changes by setting nu = 0 and all other parameters to be the same as for the SEIR model. If everything is working correctly both models should give the same output.
2. Can you alter the parameters defined above to answer the questions for this model from practical 2?

## 3. Translate a more realistic SHLIR model flow diagram to equations

As a more advanced exercise (*feel free to skip this if wanting to design your own model now*) we now translate the more complex SHLIR model into code. Look back at practical 2 for a refresher on the structure of this model.

### Populations and Initialisation

The new populations have been added for you. The population has now been split between low and high risk populations, with the only infectious case being in the high risk population.

real\_SHLIR\_inits = c(  
 # General population  
 S = 800,  
 H = 0,  
 L = 0,  
 I = 0,  
 Tr = 0,  
 R = 0,  
 ## High risk population  
 S\_H = 199,  
 H\_H = 0,  
 L\_H = 0,  
 I\_H = 1,  
 Tr\_H = 0,  
 R\_H = 0  
)

### Parameters

The required model parameters have been defined for you (with the units being years-1). The only new parameters are the between group mixing (M), the proportion that are born high risk (p), and the transmission probability in those considered to be high risk (beta\_H).

real\_SHLIR\_parameters <- c(  
 beta = 3, # Rate of transmission  
 beta\_H = 6, # High risk rate of transmission  
 gamma\_H = 1/5, # Rate of progression to active symptoms from high risk latent  
 nu = 1/2, #Rate of progression from high to low risk latent  
 gamma\_L = 1/100, # Rate of progression to active symptoms for low risk latent  
 epsilon = 1/3, # Rate of treatment  
 tau = 1/2, # Rate of recovery  
 mu = 1/81, # Rate of natural mortality  
 p = 0.2, # proportion of new births that are high risk  
 M = 0.2 # Between group mixing  
)

### Equations

Update the model equations using the model flow diagram above. The comments in the code given hints as to where changes need to be made. The equations for the low risk population have been partially provided for you.

real\_SHLIR\_demo\_ode <- function(t, x, params) {  
  
 ## Specify model compartments   
 ## The compartments for the low risk population have been provided   
 ## Add the high risk population  
 ## Don't forget to update indexing for x. Compare the previous two models for a hint.  
 S <- x[1]  
 H <- x[2]  
 L <- x[3]  
 I <- x[4]  
 Tr <- x[5]  
 R <- x[6]  
   
 S\_H <- x[7]  
 H\_H <- x[8]  
 L\_H <- x[9]  
 I\_H <- x[10]  
 Tr\_H <- x[11]  
 R\_H <- x[12]  
  
 with(as.list(params),{  
  
 ## Specify total population  
 N = S + H + L + I + Tr + R + S\_H + H\_H + L\_H + I\_H + Tr\_H + R\_H  
  
 ## Force of infection  
 ## The force of infetion in the low risk population has been provided for you  
 foi <- beta \* I / N + M \* beta\_H \* I\_H / N   
 ## Add the high risk force of infection here  
   
 ## Derivative Expressions  
 ## General population   
 ## Compare these equations from those used for the previous models  
 dS = -S \* foi - mu \* S + (1 - p) \* mu \* N  
 dH = (S + L + R) \* foi - gamma\_H \* H - nu \* H - mu \* H  
 dL = nu \* H - L \* foi - gamma\_L \* L - mu \* L  
 dI =  
 dTr = epsilon \* I - tau \* Tr - mu \* Tr  
 dR = tau \* Tr - R \* foi - mu \* R  
   
 ## High risk population  
 ## Copy the equations used above for the low risk population  
 ## Convert them to be in the high risk population  
  
   
   
   
   
   
   
  
 ## output  
 derivatives <- c(dS, dH, dL, dI, dTr, dR, dS\_H, dH\_H, dL\_H, dI\_H, dTr\_H, dR\_H)  
  
 list(derivatives)  
 })  
}

### Simulate and Summarise

Simulate the new model as previously.

begin\_time <- 0  
end\_time <- 200  
real\_SHLIR\_times <- seq(begin\_time, end\_time, 1)  
  
## see ?solve\_ode for details  
real\_SHLIR\_sim <- biddmodellingcourse::solve\_ode(model = real\_SHLIR\_demo\_ode,   
 inits = real\_SHLIR\_inits,   
 params = real\_SHLIR\_parameters,  
 times = real\_SHLIR\_times,   
 as.data.frame = TRUE)  
real\_SHLIR\_sim %>%   
 head %>%   
 pretty\_table(caption = "First 5 years of model simulation",  
 label = "tab-real\_SHLIR")

biddmodellingcourse::summarise\_model(real\_SHLIR\_sim) %>%   
 pretty\_table(caption = "Realistic SHLIR model summary statistics; The final size of each population at the end of the simulation, along with the time the epidemic peak was reached, the number of infected at the epidemic peak and the duration of the epidemic")

## For an interactive graph change interactive to TRUE  
biddmodellingcourse::plot\_model(real\_SHLIR\_sim, interactive = FALSE)

### Explore

1. Test your changes by setting all the parameters to be the same as in the SHLIR model.
2. Can you alter the parameters defined above to answer the questions for this model from practical 2?

## 4. Implement Your Own Model

Using the same structure as used in models above implement your own model into R. Talk to the instructors for tips, tricks, and potential ideas. See the idmodelr package for other example model structures (<https://www.samabbott.co.uk/idmodelr/>). It may help to first draw out the flow diagram for your model (potentially writing out the equations may also help).