Isoclines, phase plane and Ross-Macdonal

Ottar N. Bjørnstad

Phase Analysis

When working with dynamical systems one is often interested in studying the dynamics in the phase plane and derive the *isoclines* that divide this plane in regions of increase and decrease of the various state variables. The *phaseR* package is a wrapper around *ode* that makes it easy to visualize 1- and 2-dimensional differential equation flows. The \(R\) state in the SIR model does not influence the dynamics, so we can rewrite the SIR model as a 2D system.

```
require(epimdr2)
```

```
## Loading required package: epimdr2
```

```
## Loading required package: shiny
```

```
## Loading required package: deSolve
```

```
## Loading required package: plotly
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
## Loading required package: polspline
```

```
simod=function(t, y, parameters){
  S=y[1]
  I=y[2]

  beta=parameters["beta"]
  mu=parameters["mu"]
  gamma=parameters["gamma"]
  N=parameters["N"]

  dS = mu * (N  - S)  - beta * S * I / N
  dI = beta * S * I / N - (mu + gamma) * I
  res=c(dS, dI)
  list(res)
}
```

The isoclines (sometimes called the null-clines) in this system are given by the solution to the equations \(dS/dt=0\) and \(dI/dt=0\) and partitions the phase plane into regions were \(S\) and \(I\) are increasing and decreasing. For \(N=1\), the \(I\)-isocline is \(S = (\gamma +\mu)/\beta = 1/R_0\) and the S-isocline is \(I= \mu (1/S-1)/\beta\). We can draw these in the phase plane and add a simulated trajectory to the plot in a counter-clockwise dampened fashion towards the endemic equilibrium. To visualize the expected change to the system at arbitrary points in the phase plane, we can further use the function flowField in the phaseR package to superimpose predicted arrows of change.

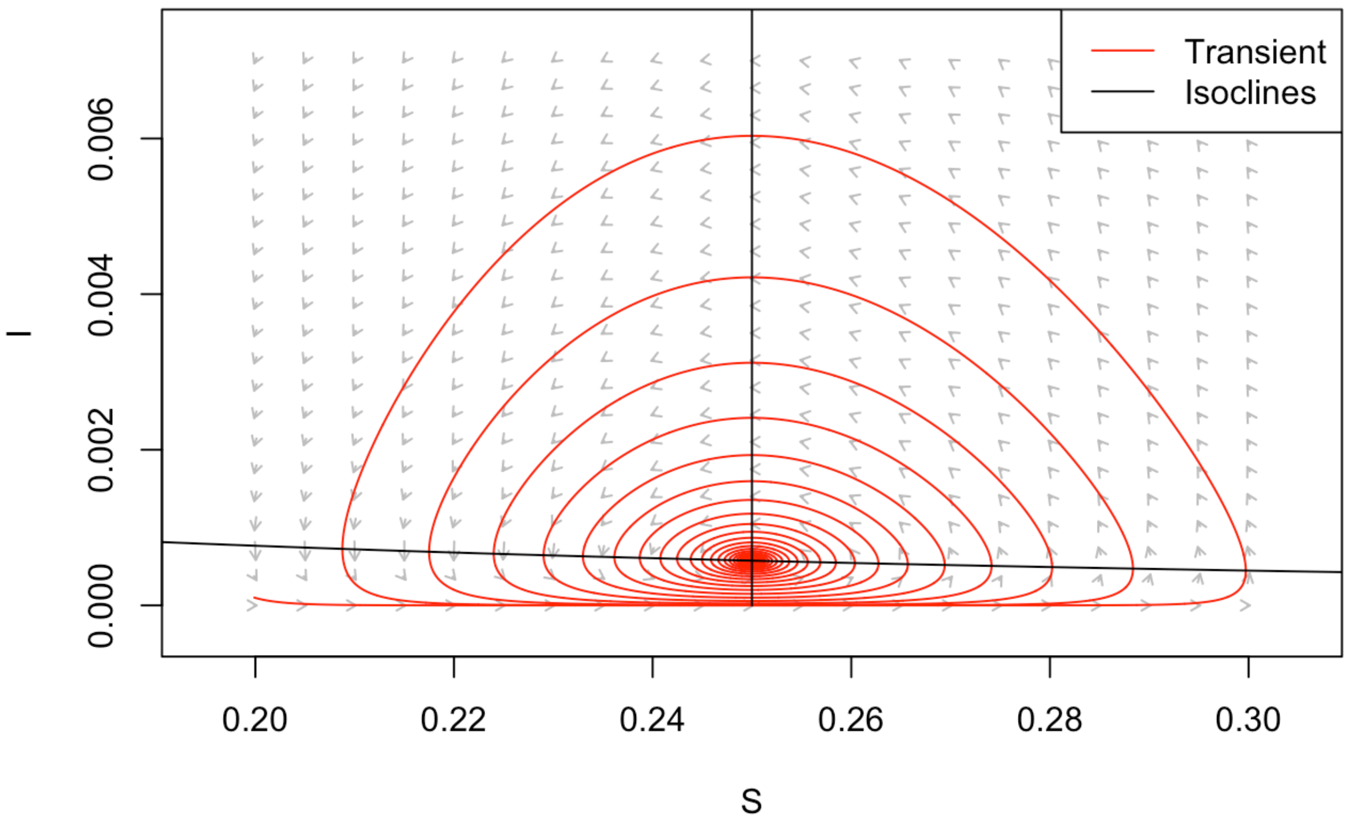
```
#parameters etc
times  = seq(0, 50, by=1/365)
 paras  = c(mu = 1/50, N = 1, R0=4, gamma = 365/14)
 paras["beta"]=paras["R0"]*(paras["gamma"]+paras["mu"])
 start = c(S=0.1999, I=0.0001, R = 0.8)*paras["N"]

require(phaseR)
```

```
## Loading required package: phaseR
```

```
## -------------------------------------------------------------------------------
## phaseR: Phase plane analysis of one- and two-dimensional autonomous ODE systems
## -------------------------------------------------------------------------------
##
## v.2.1: For an overview of the package's functionality enter: ?phaseR
##
## For news on the latest updates enter: news(package = "phaseR")
```

```
#Plot vector field
fld = flowField(simod, xlim = c(0.2,0.3), ylim = c(0,.007),
     parameters = paras, system = "two.dim",
     add = FALSE, ylab = "I", xlab = "S")
#Add trajectory
out = as.data.frame(ode(y = c(S = 0.1999, I = 0.0001),
     times =  seq(0, 52*100, by = 1/365), func = simod,
     parms = paras))
 lines(out$S, out$I, col = "red")
#Add S-isocline
curve(paras["mu"]*(1/x-1)/paras["beta"], 0.15, 0.35,
     xlab = "S", ylab = "I", add = TRUE)
#Add I-isocline
icline = (paras["gamma"] + paras["mu"])/paras["beta"]
lines(rep(icline, 2), c(0,0.01))
legend("topright", legend = c("Transient", "Isoclines"),
     lty = c(1, 1), col = c("red", "black"))
```



Stability and Periodicity

For continuous-time ODE models like the SIR, equilibria are locally stable if (and only if) all the real part of the eigenvalues of the when evaluated at the equilibrium are smaller than zero. An equilibrium is (i) a node (all trajectories moves monotonically towards/away from the equilibrium) if the largest eigenvalue has only a real part and (ii) a focus (trajectories spiral towards or away from the equilibrium) if the largest eigenvalues are a conjugate pair of complex numbers (\(a \pm b \imath\)). For a focus the imaginary part determines the dampening period of the cycle according to \(2 \pi / b\). We can thus use the Jacobian matrix to study the SIR model's equilibria. If we set \(F = dS/dt = \mu (N - S) - \beta S I / N\) and \(G = dI/dt = \beta S I / N - (\mu + \gamma) I\), the Jacobian of the SIR system is \[\begin{equation} \vec{J}=\left(\begin{array}{cc} \frac{\partial F}{\partial S} & \frac{\partial F}{\partial I} \\ \frac{\partial G}{\partial S} & \frac{\partial G}{\partial I} \end{array} \right), \end{equation}\] and the two equilibria are the disease free equilibrium and the endemic equilibrium as defined above.

can help with all of this. The endemic equilibrium is:

```
#Pull values from paras vector
gamma = paras["gamma"]
beta = paras["beta"]
mu = paras["mu"]
N = paras["N"]
#Endemic equilibrium
Sstar=(gamma+mu)/beta
Istar=mu*(beta/(gamma+mu)-1)/beta
eq1=list(S=Sstar, I=Istar)
```

The elements of the Jacobian using ’s differentiation function are

```
#Define equations
dS = quote(mu * (N  - S)  - beta * S * I / N)
dI = quote(beta * S * I / N - (mu + gamma) * I)
#Differentiate w.r.t. S and I
j11 = D(dS, "S")
j12 = D(dS, "I")
j21 = D(dI, "S")
j22 = D(dI, "I")
```

Pass the values for \(S^*\) and \(I^*\) in the *eq1* list to the Jacobian and use the *eigen* function to calculate the eigenvalues:

```
#Evaluate Jacobian at equilibrium
JJ=with(data=eq1, expr=matrix(c(eval(j11),eval(j12),
     eval(j21),eval(j22)), nrow=2, byrow=TRUE))
#Calculate eigenvalues
eigen(JJ)$values
```

```
## [1] -0.04+1.250554i -0.04-1.250554i
```

For the endemic equilibrium, the eigenvalues is a pair of complex conjugates which real parts are negative, so it is a stable focus. The period of the inwards spiral is:

```
2*pi/(Im(eigen(JJ)$values[1]))
```

```
## [1] 5.024321
```

So with these parameters the dampening period is predicted to be just over 5 years. Thus, during disease invasion we expect this system to exhibit initial outbreaks every 5 years. A further significance of this number is that if the system is stochastically perturbed by environmental variability affecting transmission, the system will exhibit low amplitude ``phase-forgetting” cycles with approximately this period in the long run.

The same protocol can be used for the disease free equilibrium $\{S^*=1, I^*=0\}$.

```
eq2=list(S=1,I=0)
JJ=with(eq2,
     matrix(c(eval(j11),eval(j12),eval(j21),
     eval(j22)), nrow=2, byrow=TRUE))
eigen(JJ)$values
```

```
## [1] 78.27429 -0.02000
```

The eigenvalues are strictly real and the largest value is greater than zero, so it is an unstable node (a ``saddle”); The epidemic trajectory is predicted to move monotonically away from this disease free equilibrium if infection is introduced into the system. This makes sense because with the parameter values used, $R_0 = 4$ which is greater than the invasion threshold value of 1.