Phase space and vector field in R (for two-dimensional system)

Rudolf P. Rohr

4/16/2020

Phase space

Let us consider again the following example of a two-dimensional system:

$$\frac{dN_1}{dt} = N_1 \cdot r - \alpha N_1^2 - \beta \cdot N_1 \cdot N_2$$

$$\frac{dN_2}{dt} = +\beta N_1 \cdot N_2$$

Load the library deSolve:

```
library(deSolve)
```

The list of parameters:

```
p <- list(r = 0.2, alpha = 0.1, beta = 0.2)
```

The differential equation function:

```
f <- function(t,N,p){
  N1 <- N[1]
  N2 <- N[2]
  dN1 <- p$r * N1 - p$alpha * N1 * N1 - p$beta * N1 * N2
  dN2 <- + p$beta * N1 * N2
  return(list(c(dN1,dN2)))
}
```

The time steps

```
time_steps <- seq(0,50,0.05)
```

The initial condition:

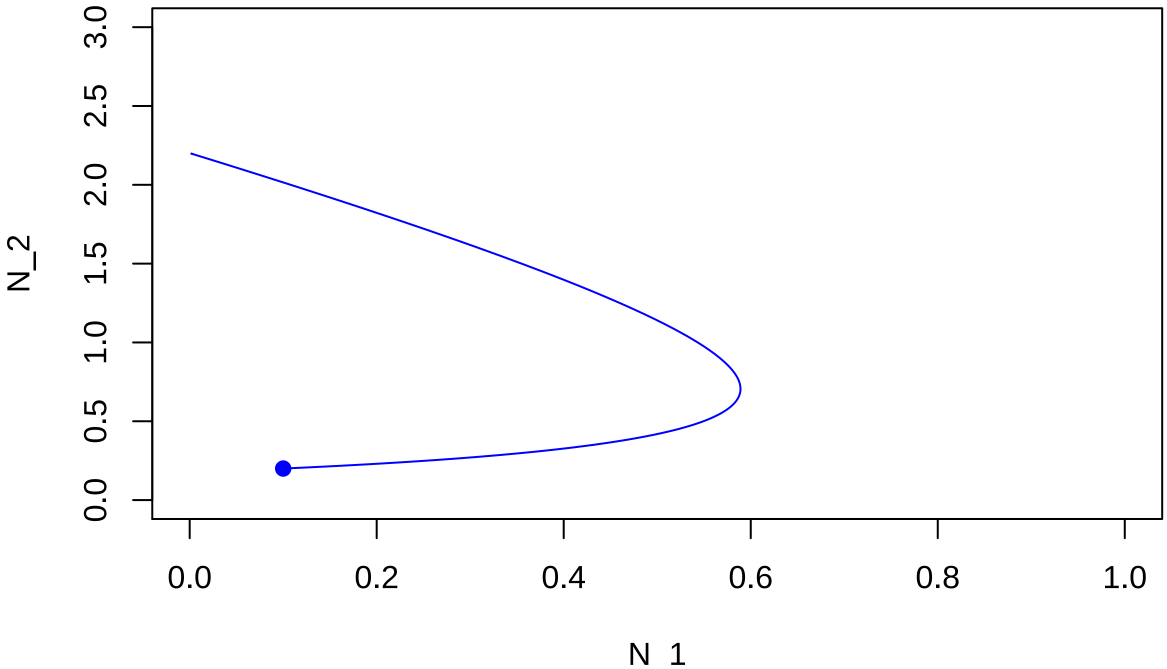
```
N0 <- c(0.1, 0.2)
```

Running the "ode" function:

```
out <- ode(y = N0, times = time_steps, func = f, parms = p, method = c("ode45"))
```

Finally, we can plot the phase space, which represents the trajectory in the N_1-N_2 space :

```
plot(NA, xlab = "N_1", ylab="N_2", xlim=c(0,1), ylim=c(0,3))
lines(x = out[,2], y = out[,3], col='blue')
points(N0[1],N0[2],pch=19,cex=1,col='blue')
```



The blue dot represents the initial condition $N0$.

Then, we can also represent several trajectories with different initial conditions on the same phase space. This gives us an idea of the behaviour of the dynamical system. Here, we chose 6 different initial conditions:

```
N1 <- c(0.5, 2)
N2 <- c(0.5, 1)
N3 <- c(0.5, 0.5)
N4 <- c(0.8, 2)
N5 <- c(0.8, 1)
N6 <- c(0.8, 0.5)
```

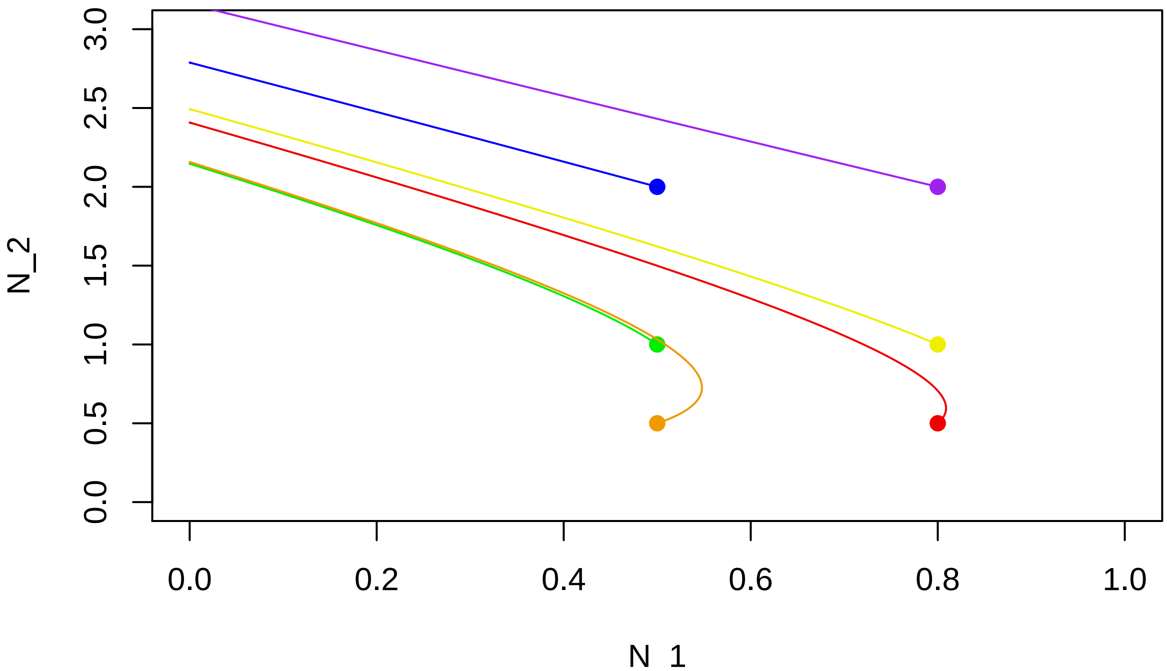
Solve the system for those 6 initial conditions:

```
out1 <- ode(y = N1, times = time_steps, func = f, parms = p, method = c("ode45"))
out2 <- ode(y = N2, times = time_steps, func = f, parms = p, method = c("ode45"))
out3 <- ode(y = N3, times = time_steps, func = f, parms = p, method = c("ode45"))
out4 <- ode(y = N4, times = time_steps, func = f, parms = p, method = c("ode45"))
out5 <- ode(y = N5, times = time_steps, func = f, parms = p, method = c("ode45"))
out6 <- ode(y = N6, times = time_steps, func = f, parms = p, method = c("ode45"))
```

Finally, plot the phase space with those 6 trajectories:

```
plot(NA, xlab = "N_1", ylab="N_2", xlim=c(0,1), ylim=c(0,3))
lines(x = out1[,2], y = out1[,3], col='blue')
points(N1[1],N1[2],pch=19,cex=1,col='blue')
lines(x = out2[,2], y = out2[,3], col='green2')
points(N2[1],N2[2],pch=19,cex=1,col='green2')
lines(x = out3[,2], y = out3[,3], col='orange2')
points(N3[1],N3[2],pch=19,cex=1,col='orange2')
lines(x = out4[,2], y = out4[,3], col='purple')
points(N4[1],N4[2],pch=19,cex=1,col='purple')
lines(x = out5[,2], y = out5[,3], col='yellow2')
points(N5[1],N5[2],pch=19,cex=1,col='yellow2')
lines(x = out6[,2], y = out6[,3], col='red2')
```

```r
points(N6[1],N6[2],pch=19,cex=1,col='red2')
```



From this phase space, we may conclude that the trajectories converge to $N_1 = 0$ and to a point $N_2 > 0$ which depends on the initial conditions (like for S in the SI model).

Vector field

To study numerically the trajectoires of a two-dimensional system, we could draw trajectoires from substantial set of initial conditions. But we can also draw the vector field of the system. For each point (N_1, N_2) the time derivative

$$\frac{dN_1}{dt} = N_1 \cdot r - \alpha N_1^2 - \beta \cdot N_1 \cdot N_2$$

$$\frac{dN_2}{dt} = +\beta N_1 \cdot N_2$$

give the direction of the trajectory passing by this point. That is, for each point (N_1, N_2), we can define the vector

$$\vec{v} = \begin{bmatrix} N_1 \cdot r - \alpha N_1^2 - \beta \cdot N_1 \cdot N_2 \\ +\beta N_1 \cdot N_2 \end{bmatrix}$$

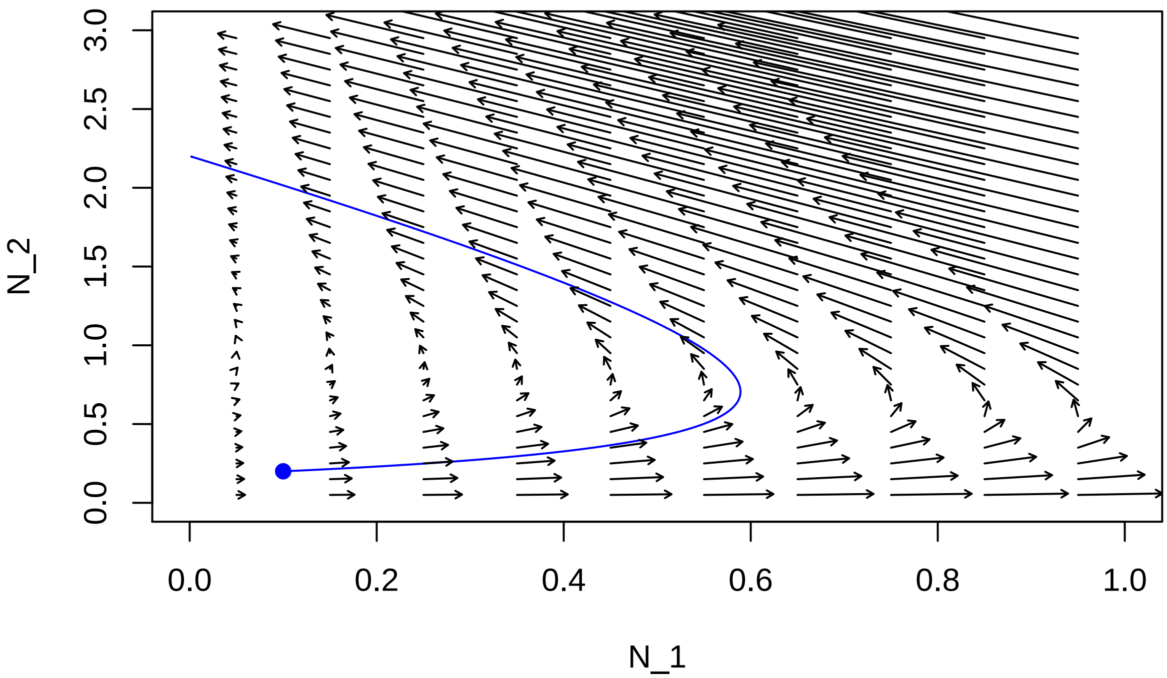
This set of vectors are tangent to the trajectories. Therfore, by drawing these vectors for a dense subset of points (N_1, N_2) one can "see" all trajectories at once. This can be done in the following way. First we define the interval on the N_1 and N_2 axes at which we want to draw a vector.

```r
N1.g <- seq(0.05,0.95,0.1)
N2.g <- seq(0.05,2.95,0.1)
```

Then, using a double loop we can draw the vector field on the phase space.

```r
plot(NA, xlab = "N_1", ylab="N_2", xlim=c(0,1), ylim=c(0,3))
lines(x = out[,2], y = out[,3], col='blue')
points(N0[1],N0[2],pch=19,cex=1,col='blue')
for (i in 1:length(N1.g)){
  for (j in 1:length(N2.g)){
```

```
    N1 <- N1.g[i]
    N2 <- N2.g[j]
    dN1 <- p$r * N1 - p$alpha * N1 * N1 - p$beta * N1 * N2
    dN2 <- + p$beta * N1 * N2
    arrows(N1, N2, N1+dN1, N2+dN2, length = 0.04)
  }
}
```



Here, the trajectory with initial condition $N0$ is represented in blue. As an exercise, you can add the six other trajectories drawn above.