

# Lab 3

*Clint Valentine*

## Modeling the Dynamics of Structured Populations

We will use the Lotka-Volterra model for direct competition between two species as described below in the system of differential equations describing  $N_1$  and  $N_2$ :

$$1.1 \quad \frac{dN_1}{dt} = r_1 N_1 \left( \frac{K_1 - N_1 - \alpha_{12} N_2}{K_1} \right)$$

$$1.2 \quad \frac{dN_2}{dt} = r_2 N_2 \left( \frac{K_2 - N_2 - \alpha_{21} N_1}{K_2} \right)$$

This system of differential equations is easily solvable using the `ode45` algorithm. An example system is proposed, solved, and plotted.

```
library(deSolve)
```

```
## Warning: package 'deSolve' was built under R version 3.1.3
```

```
library(manipulate)
library(fields)
```

```
## Warning: package 'fields' was built under R version 3.1.3
```

```
## Loading required package: spam
```

```
## Warning: package 'spam' was built under R version 3.1.3
```

```
## Loading required package: grid
## Spam version 1.0-1 (2014-09-09) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
##
## Attaching package: 'spam'
##
## The following objects are masked from 'package:base':
##
##      backsolve, forwardsolve
##
## Loading required package: maps
```

```
## Warning: package 'maps' was built under R version 3.1.3
```

```

source('http://faraway.neu.edu/data/cb.R')

solve.lotka.volterra <- function(t, y, parms) {
  dX <- parms$r1 * y[1] * ((parms$K1 - y[1] - parms$alpha12 * y[2]))
  dY <- parms$r2 * y[2] * ((parms$K2 - y[2] - parms$alpha21 * y[1]))
  return(list(c(dX, dY)))
}

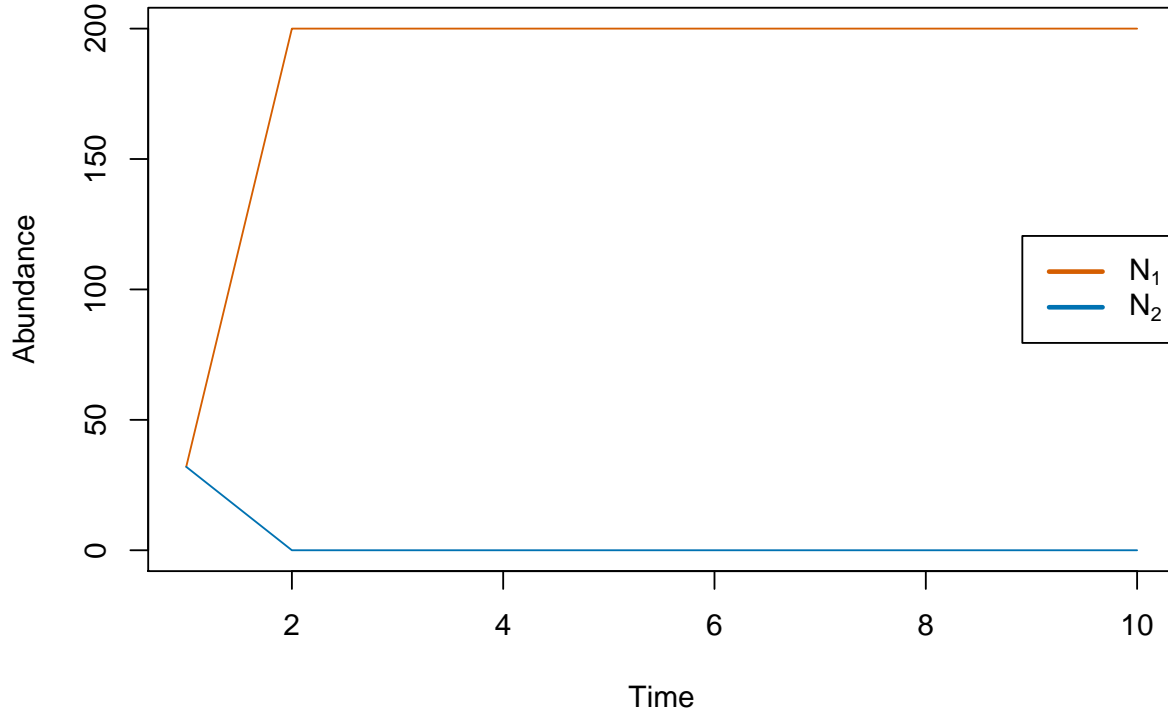
N = c(32, 32)
parms = list(K1=200, K2=110, r1=2, r2=2, alpha12=2, alpha21=2)
times <- 1:10

results <- ode(y=N, times=times, func=solve.lotka.volterra, parms,
               method="ode45")

matplot(results[,2:3], type='l', lty=1, xlab='Time', ylab='Abundance',
         col=c(cb.red, cb.blue))

legend(x='right', legend=c(expression('N'[1]), expression('N'[2])),
       lty=c(1,1), lwd=c(2.5,2.5), col=c(cb.red, cb.blue))

```



To graphically display the phase diagram for a dual species Lotka-Volterra model we will first define the zero net growth isoclines of the system. We will then simulate many starting points of the system and draw their solutions as lines on the phase diagram.

```

plot.competition.ZNGI <- function(grid, time.length,
                                   K1, K2, r1, r2, alpha12, alpha21) {

  parms <- list(K1=K1, K2=K2, r1=r1, r2=r2, alpha12=alpha12, alpha21=alpha21)
  times <- seq(0, 1, len=time.length)

  N1 <- matrix(
    c(0, (parms$K1 / parms$alpha12), parms$K1, 0), nrow=2, ncol=2, byrow=TRUE)
  N2 <- matrix(
    c(0, parms$K2, (parms$K2 / parms$alpha21), 0), nrow=2, ncol=2, byrow=TRUE)

  xmax <- max(parms$K1, parms$K2 / parms$alpha21) + 1
  ymax <- max(parms$K2, parms$K1 / parms$alpha12) + 1

  x.equil <- (parms$K1 - parms$alpha12 * parms$K2) /
    (1 - parms$alpha12 * parms$alpha21)
  y.equil <- (parms$K2 - parms$alpha21 * parms$K1) /
    (1 - parms$alpha12 * parms$alpha21)

  N1.vect = seq(0.001, max(parms$K1, parms$K2 / parms$alpha21), len=grid)
  N2.vect = seq(0.001, max(parms$K2, parms$K1 / parms$alpha12), len=grid)

  initials <- expand.grid(N1.vect, N2.vect)

  plot(NULL, xlim=c(0,xmax), ylim=c(0,ymax), xlab=expression('N'[1]),
        ylab=expression('N'[2]), main='Zero Net Growth Isoclines for Two Species')

  for (x in 1:length(initials[,1])) {
    N <- c(initials[x, 1], initials[x, 2])
    results <- ode(y=N, times=times, func=solve.lotka.volterra, parms,
                  method='ode45')
    lines(results[,2], results[,3], type='l', lty=1)
  }

  lines(N1, type='l', lty=1, lwd=3, col=cb.blue)
  lines(N2, type='l', lty=1, lwd=3, col=cb.red)

  if (N1[1, 2] < N2[1, 2]) {
    y.boundary <- N2[1, 2]
    points(0, y.boundary, col=cb.yellow, pch=16)
    points(x.equil, y.equil, col=cb.yellow, pch='o')
  } else if (N1[1, 2] > N2[1, 2]) {
    y.boundary <- N1[1, 2]
    points(0, y.boundary, col=cb.yellow, pch='o')
    points(x.equil, y.equil, col=cb.yellow, pch=16)
  }

  if (N1[2, 1] < N2[2, 1]) {
    x.boundary <- N2[2, 1]
    points(x.boundary, 0, col=cb.yellow, pch='o')
  } else if (N1[2, 1] > N2[2, 1]) {
    x.boundary <- N1[2, 1]

```

```

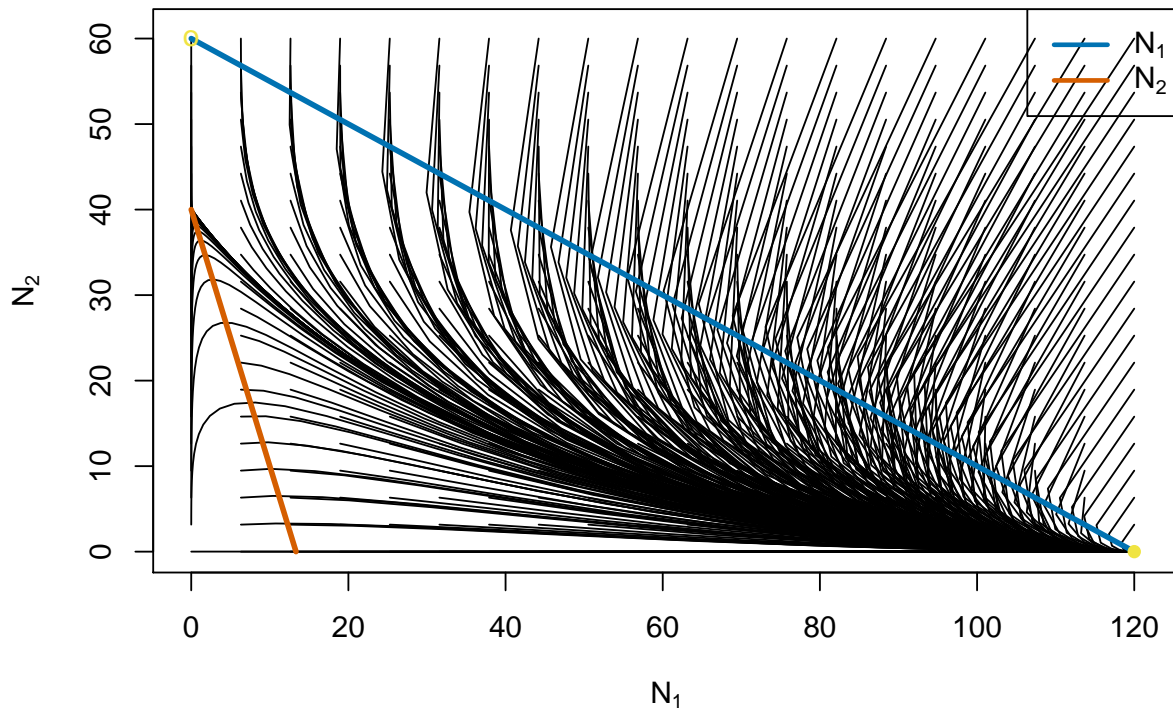
    points(x.boundary, 0, col=cb.yellow, pch=16)
  }

  legend(x='topright', legend=c(expression('N'[1]), expression('N'[2])),
        lty=c(1,1), lwd=c(2.5,2.5), col=c(cb.blue, cb.red))
}

plot.competition.ZNGI(grid=20, time.length=100, 120, 40, 0.5, 0.35, 2, 3)

```

## Zero Net Growth Isoclines for Two Species



The following code was used to manipulate the zero net growth isocline plotting function in real time:

```

r manipulate ({ plot.competition.ZNGI(gridVal, Tlen, K1Val, K2Val, r1Val, r2Val, alpha12Val,
alpha21Val) }, gridVal=slider(min=5, max=100, initial=20, label="Grid Dimension"), Tlen=slider(min=100,
max=10000, initial=100, label="Timestep Resolution"), K1Val=slider(min=20, max=500,
initial=100, label="K1 Value"), K2Val=slider(min=20, max=500, initial=100, label="K2
Value"), r1Val=slider(min=0, max=4, initial=0.35, step=0.05, label="Rate for N1"), r2Val=slider(min=0,
max=4, initial=0.35, step=0.05, label="Rate for N2"), alpha12Val=slider(min=0, max=30,
initial=2, label="Alpha12"), alpha21Val=slider(min=0, max=30, initial=2, label="Alpha21")
)

```

## Stability Analysis

An eigen analysis of the Jacobian matrix of the five potential equilibrium points will also elucidate the behaviour of the Lotka-Volterra interaction. First, the three boundary conditions are tested. The extinction condition at  $N_1 = 0$  and  $N_2 = 0$  is stable only if  $r_1$  and  $r_2$  are greater than zero. Either monoculture boundary

condition are stable if the species growth rate is positive and if the ratio of the competitor's carrying capacity to the species' carrying capacity is smaller than the competitive effect of the species on the competitor.

The interior equilibrium is only present if it exists in the domain of biological relevance (positive integers). This point is stable if the intraspecific competition of both species is greater than the interspecific species.

If both species have stable monoculture points than there are two alternative stable equilibriums in the system with a potential unstable coexistence point.

The following functions determine the characteristics of a system graphically and numerically using eigen analysis of the Jacobian matrix.

```
make.Jacob <- function(N1, N2, K1, K2, r1, r2, alpha12, alpha21) {
  J <- matrix(
    data=c(
      (r1 / K1) * (K1 - (2 * N1) - (alpha12 * N2)),
      (-r1 / K1) * (alpha12 * N1),
      (-r2 / K2) * (alpha21 * N2),
      (r2 / K2) * (K2 - (2 * N2) - (alpha21 * N1))),
    nrow=2, ncol=2, byrow=FALSE)
  return(J)
}

test.Stability <- function(K1, K2, r1, r2, alpha12, alpha21) {

  b.origin <- Re(eigen(make.Jacob(0, 0, K1, K2, r1, r2, alpha12, alpha21))$values)

  if (b.origin[1] < 0 && b.origin[2] < 0) {
    return(1)
  }

  if (alpha12 * alpha21 != 1) {
    N1.equil <- (K1 - alpha12 * K2) / (1 - alpha12 * alpha21)
    N2.equil <- (K2 - alpha21 * K1) / (1 - alpha12 * alpha21)
    if (N1.equil > 0 && N2.equil > 0) {
      i.equil <- Re(eigen(make.Jacob(N1.equil, N2.equil, K1, K2, r1, r2,
        alpha12, alpha21))$values)
      if (i.equil[1] < 0 && i.equil[2] < 0) {
        return(4)
      }
    }
  }

  b.N1 <- Re(eigen(make.Jacob(K1, 0, K1, K2, r1, r2, alpha12, alpha21))$values)
  b.N2 <- Re(eigen(make.Jacob(0, K2, K1, K2, r1, r2, alpha12, alpha21))$values)

  N1.mono <- (b.N1[1] < 0 & b.N1[2] < 0)
  N2.mono <- (b.N2[1] < 0 & b.N2[2] < 0)

  if (N1.mono & N2.mono) {
    return(5)
  } else if (N1.mono) {
    return(2)
  } else if (N2.mono) {
    return(3)
  }
}
```

```

}

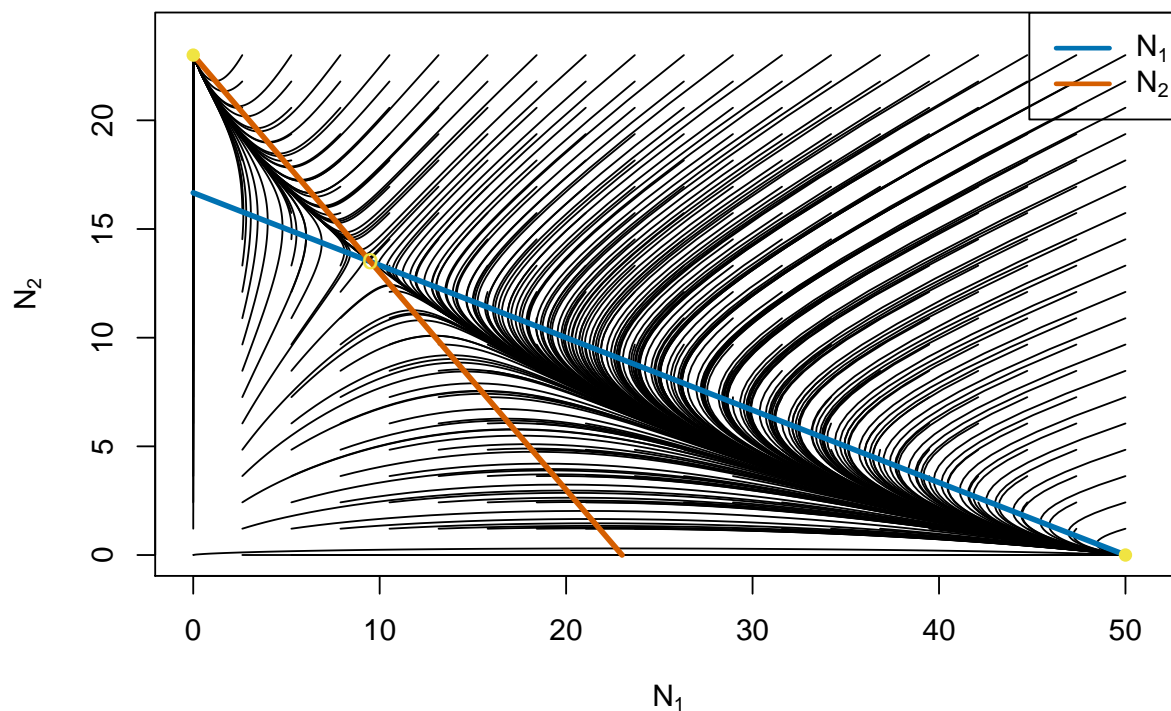
test.Stability(50, 23, 3, 4, 3, 1) # 5 = Multiple Stable States

## [1] 5

plot.competition.ZNGI(20, 1000, 50, 23, 3, 4, 3, 1)

```

## Zero Net Growth Isoclines for Two Species



We will now simulate this system for all variations of the following parameter values:  $\alpha_{12} = 0.5$ ,  $\alpha_{21}$  ranging from 0 to 2.5,  $K_1 = 3$ ,  $K_2$  ranging from 0.01 to 15,  $r_1 = r_2 = 2$ . These results are then stored into a matrix for future plotting.

```

source("http://faraway.neu.edu/data/format3d.R")

p <- expand.grid(alpha12=0.5, alpha21=seq(0, 2.5, len=100), r1=2, r2=2,
                K1=3, K2=seq(0.01, 15, len=100))

results = matrix(nrow = nrow(p), ncol=3)

for (x in 1:nrow(p)) {
  pr <- p[x,]
  results[x,] <- cbind(pr$K2/pr$K1,
                      pr$alpha21/pr$alpha12,
                      test.Stability(pr$K1, pr$K2, pr$r1, pr$r2,

```

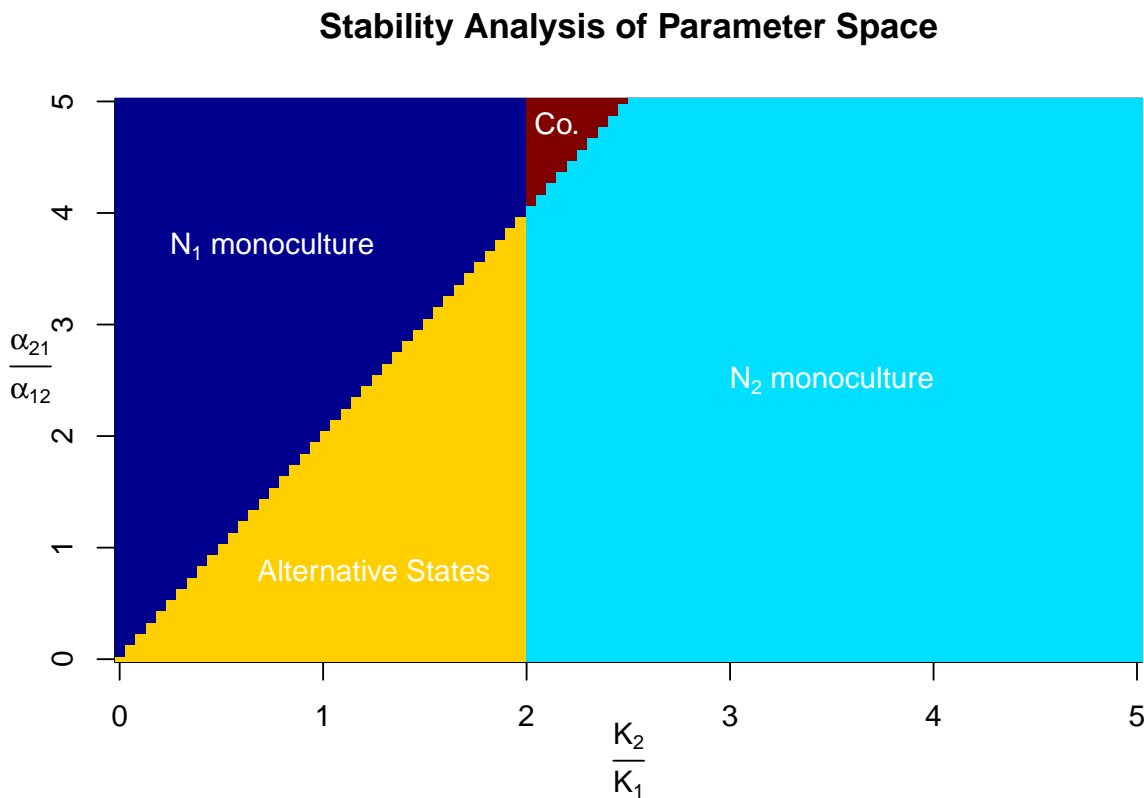
```

    pr$alpha12, pr$alpha21))
}

results.3d <- format3d(results)
image(results.3d, xlab=expression(frac(K[2], K[1])), main='Stability Analysis of Parameter Space', col=

text(x=-0.25, y=2.5, expression(frac(alpha[21], alpha[12])), xpd=TRUE, srt=0, pos=2)
text(0.75, 3.7, col='white', expression(paste(N[1], ' monoculture'))))
text(1.25, 0.8, col='white', 'Alternative States')
text(2.15, 4.8, col='white', 'Co.')
text(3.5, 2.5, col='white', expression(paste(N[2], ' monoculture'))))

```



This plot makes sense. I ran a few simulations with parameter values in this space and I saw the corresponding zero net growth isoclines indicative of these scenarios. When  $K_1$  is larger than the value of  $K_1$  there are two outcomes determined by the ratio of the species' respective interspecific competitive effect. We see an  $N_1$  monoculture when  $\alpha_{21} > \alpha_{12}$  and alternative monocultures of the two species when  $\alpha_{21} < \alpha_{12}$  depending on the initial population values.

When  $K_2$  is larger than the value of  $K_1$  there are also two outcomes that are determined by the ratio of the species' respective interspecific competitive effect. When  $\alpha_{21} > \alpha_{12}$  we see coexistence of the two species and when  $\alpha_{21} < \alpha_{12}$  we see a complete  $N_2$  monoculture.

## Extending and Simulating Mechanistic Models of Competition

To extend the model above to mechanistic competition by modeling the common resource between the two species we must use a system of three differential equations. Instead of using parameters that simulate a direct competitive effect on each species we will look at indirect competition through a common essential resource.

These are not right at all

$$1.1 \quad \frac{dN_1}{dt} = r_1 N_1 \left( \frac{K_1 - N_1 - \alpha_{12} N_2}{K_1} \right)$$

$$1.2 \quad \frac{dN_2}{dt} = r_2 N_2 \left( \frac{K_2 - N_2 - \alpha_{21} N_1}{K_2} \right)$$

```
library(deSolve)

solve.mechanistic <- function(t, y, parms) {
  dR <- parms$r1 * (parms$S1 - y[1]) - (y[2] * parms$u1 * y[1]) / (parms$Y1 * (parms$K1 + y[1])) - (y[3] *
  dX <- y[2] * (parms$u1 * (y[1] / (parms$K1 + y[1])) - parms$M1)
  dY <- y[3] * (parms$u2 * (y[1] / (parms$K2 + y[1])) - parms$M2)
  return(list(c(dR, dX, dY)))
}

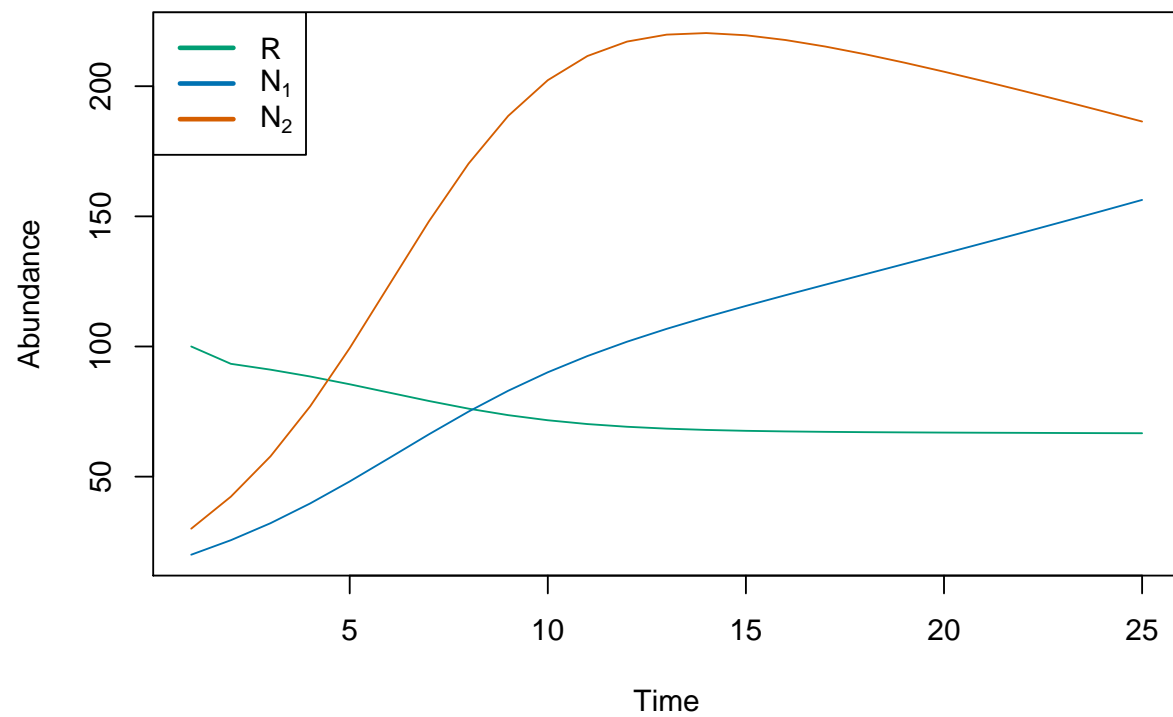
N = c(100, 20, 30)
parms = list(r1=4, S1=100, K1=32, u1=3, M1=2, K2=32, u2=5, M2=3.4, Y1=5, Y2=9)
times <- 1:25

results <- ode(y=N, times=times, func=solve.mechanistic, parms,
               method="ode45")

matplot(results[,2:4], type='l', lty=1, xlab='Time', ylab='Abundance',
         col=c(cb.green, cb.blue, cb.red))

legend(x='topleft', legend=c('R', expression('N'[1]), expression('N'[2])),
       lty=c(1,1,1), lwd=c(2.5,2.5,2.5), col=c(cb.green, cb.blue, cb.red))
```





```
R1.critical <- (parms$K1 * parms$M1) / (parms$u1 - parms$M1)
R2.critical <- (parms$K2 * parms$M2) / (parms$u2 - parms$M2)

print(R1.critical)
```

```
## [1] 64
```

```
print(R2.critical)
```

```
## [1] 68
```