

Ordinary Differential Equations in **R**

M. Hank H. Stevens

Department of Botany
Miami University
Oxford, Ohio, USA

Introduction

Using lsoda

Demo

Acknowledgements

- Stephen Cox
- Thomas Petzoldt (`simecol` package)
- Steve Ellner.

Ecology is as Simple As ...

$$\frac{dP}{dt} = \beta_{PB}\alpha_{BP}PB + \beta_{PN}\alpha_{NP}PN - m_P P \quad (1)$$

$$\frac{dN}{dt} = \beta_{NB}\alpha_{BN}BN - m_N N - \alpha_{NP}PN \quad (2)$$

$$\frac{dB}{dt} = rB(1 - \alpha_{BB}B) - \alpha_{BN}BN - \alpha_{BP}PB \quad (3)$$

ODE Solver in R

Package `odesolve`

Function `lsoda`

Additional Package `simecol` (T. Petzoldt)

lsoda

Usage `lsoda(y, times, func, parms, ...)`

Arguments Several, including,

y the *initial* values for the ODE system. If these are named, the names will label the output.

times the time points for which values of *y* are desired.
`times[1] = t0`.

func a function for the ODEs (see below).

parms a *list* of parameters used by the ODEs.

A Function For lsoda

```
yp1 <- function(t, y, parms) {  
  ## t will be input, in lsoda  
  ## y is a vector of all the state variables,  
  ## at one time point.  
  ## parms is a vector or list of parameters,  
  ## preferably named.  
  
  yp1 <- <differential equation1>  
  yp2 <- <differential equation2>  
  
  ## The result is returned as a list with one component.  
  return( list( c(yp1, yp2) )  
  )  
}
```

Logistic Growth

$$dN/dt = rN(1 - \alpha N)$$

Logistic Growth

$$dN/dt = rN(1 - \alpha N)$$

```
p <- c(r = 1, a = 0.001)
y0 <- c(N = 10)
t <- 1:20

logGrowth <- function(t, y, p) {
  dN.dt <- p[1] * y[1] * (1 - p[2] * y[1])
  return(list(dN.dt))
}
```

Logistic Growth

$$dN/dt = rN(1 - \alpha N)$$

```
p <- c(r = 1, a = 0.001)
y0 <- c(N = 10)
t <- 1:20
logGrowth <- function(t, y, p) {
  N <- y[1]
  with(as.list(p), {
    dN.dt <- r * N * (1 - a * N)
    return(list(dN.dt))
  })
}
```

Using lsoda

```
output <- lsoda(y=y0, times=t, func=logGrowth,  
               parms=p)  
matplot(output[,1], output[,-1], type='l',  
         xlab='Time', ylab='State Variables')
```

Output value is a matrix ($nrows=length(t)$,
 $ncols=length(y)$).

Demo

- Brief primer of what we just did, plus,
- ODEs in class: “Multiple Basins of Attraction.”