Taylor Glass

BI 471/571: Population Ecology

Spring 2016

## **Homework 2:** *Single species dynamics*

Due 4/14/2016

1. Hastings Problems 4.1 & 4.3

$\boxed{4.1}$ model: $\frac{dN}{dt} = RN[1-(N/K)^{\theta}]$

a) Find equilibria + determine stability

$\frac{dN}{dt} = 0$, so $0 = RN[1-(N/K)^{\theta}]$

solve for N:

$0 = RN - RN(N/K)^{\theta}$

$\frac{RN}{RN} = \cancel{RN}(N/K)^{\theta}$

$1 = (N/K)^{\theta}$

$N = K^{1/\theta}$

when $N = 0$, unstable

$N = K$, stable

graph using $\theta = 0.5$, $\theta = 1$, $\theta = 2$

$\frac{dN/dt}{N} = R[1-(N/K)^{\theta}]$

???

$\boxed{4.3}$ $\frac{dN}{dt} = RN(N-a)[1-(N/K)]$

a) equilibrium:

$0 = RN(N-a)[1-(N/K)]$

$0 = (RN^2 - RNa)[1-(N/K)]$

$0 = RN^2 - \frac{RN^3}{K} - RNa + \frac{RN^2a}{K}$

$\frac{RN^3}{K} - \frac{RN^2a}{K} = RN^2 - RNa$

$\frac{RN^2}{K}(\cancel{N-a}) = RN(\cancel{N-a})$

$RN^2 = RNK$

$N^2 = NK$

$N = K$

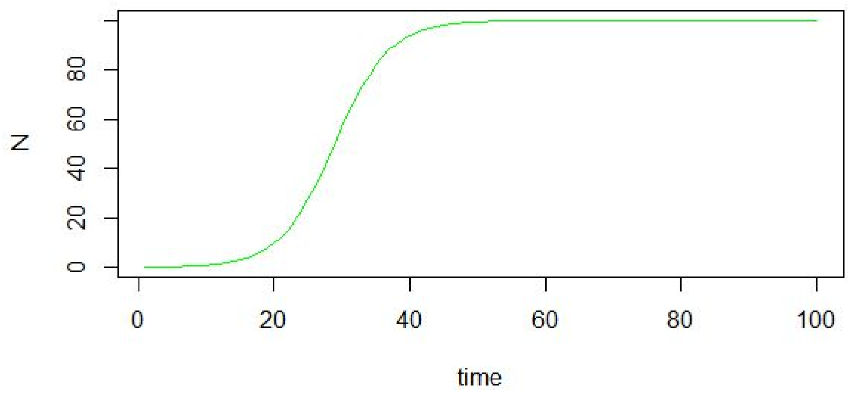
b) stability?

c) graph?

2. **(a)**

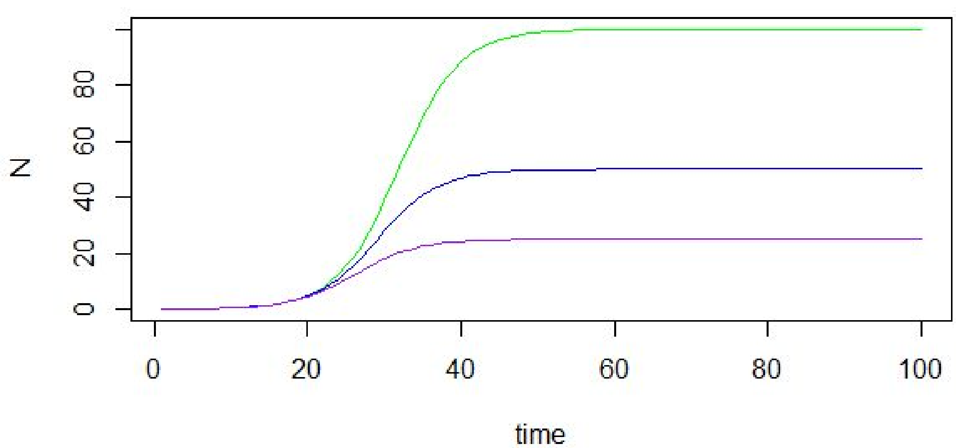
```
## write logistic growth function from exp.growth, note changes to dN.dt
log.growth <- function(t, y, p) {
  N <- y[1]
  with(as.list(p), {
    dN.dt <- r * N * (1 - (N/K))
    return(list(dN.dt))
  })
}

## make vector of parameters
p <- c('r' = 0.25, 'K' = 100)
```

```
## using runif, write for interval
0.01 to 0.1, initial conditions
y0 <- c('N' = runif(1, min = 0.01,
max = 0.1))

## time
t <- 1:100

## load library if needed
libary(deSolve)

## simulation
sim <- ode(y = y0, times = t,
func = log.growth, parms = p,
method = 'lsoda')
sim <- as.data.frame(sim)

## plot my simulation
plot(N ~ time, data = sim, type =
'l', col = 'green')
```




**(b)**

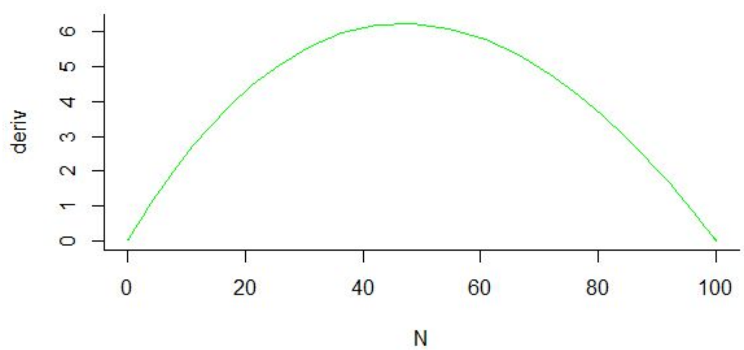
```
## write logistic growth function from exp.growth, note changes to dN.dt
log.growth <- function(t, y, p) {
  N <- y[1]
  with(as.list(p), {
    dN.dt <- r * N * (1 - (N/K))
    return(list(dN.dt))
  })
}

## make vector of
parameters
p <- c('r' = 0.25, 'K' =
100)

## using runif, write for
interval 0.01 to 0.1,
initial conditions
y0 <- c('N' = runif(1,
min = 0.01, max =
0.1))

## time
t <- 1:100

## load library if
needed
libary(deSolve)

## simulation
sim <- ode(y = y0,
times = t, func =
log.growth, parms = p,
method = 'lsoda')
sim <- as.data.frame(sim)

## plot my simulation
plot(N ~ time, data = sim, type = 'l', col = 'green')
```



```
## add plots for k = 50 and k = 25
## first for k=50 use p.2 and sim.2
p.2 <- c('r' = 0.25, 'K' = 50)
sim.2 <- ode(y = y0, times = t, func = log.growth, parms = p.2, method = 'lsoda')
sim.2 <- as.data.frame(sim.2)

## next k=25 use p.3 and sim.3
p.3 <- c('r' = 0.25, 'K' = 25)
sim.3 <- ode(y = y0, times = t, func = log.growth, parms = p.3, method = 'lsoda')
sim.3 <- as.data.frame(sim.3)

## plot on same figure, ylim will help with line
plot(N ~ time, data = sim, type ='l', col = 'green')

## using points, it allows us to use above plot and add in the other data frames
points(N ~ time, data = sim.2, type = 'l', col = 'blue')
points(N ~ time, data = sim.3, type = 'l', col = 'purple')
```

# (c)

**Code for 'r' = 0.25, 'K' = 100:**

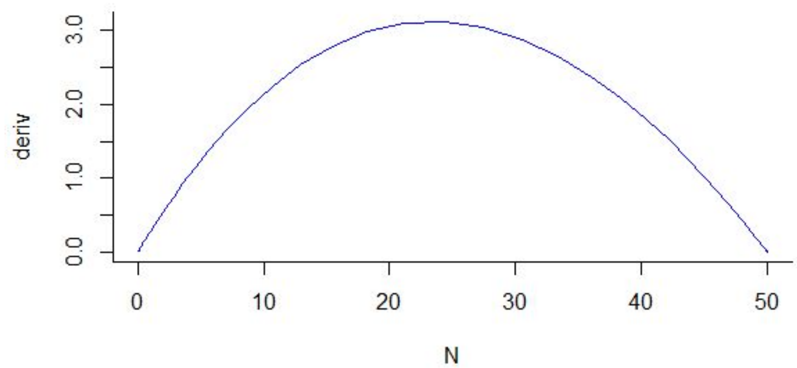



```
## theta log growth function
log.growth <- function(t, y, p) {
  N <- y[1]
  with(as.list(p), {
    dN.dt <- r * N * (1 - (N / K))
    return(list(dN.dt))
  })
}

## parameters
p <- c('r' = 0.25, 'K' = 100)
y0 <- c('N' = runif(1, min = 0.01, max =
0.1))
t <- 1:100

## simulation
sim <- ode(y = y0, times = t, func =
log.growth, parms = p, method =
'lsoda')
sim <- as.data.frame(sim)

## plot
plot(N ~ time, data = sim, type = 'l', lwd = 2, bty = 'l', col = 'green')

## compute derivatives
sim$deriv <- c(diff(sim$N), NA)

## plot vs pop abundance
plot(deriv ~ N, data = sim, type = 'l', col = 'blue', bty = 'l')

## max value
max(sim$deriv, na.rm = TRUE)
## Output: 6.235844
which(sim$deriv == max(sim$deriv, na.rm = TRUE))
## Output: 30

## value of N in sim with the same index as the biggest deriv value
```

```
sim$N[which(sim$deriv == max(sim$deriv, na.rm = TRUE))]
## Output: 48.43107
```

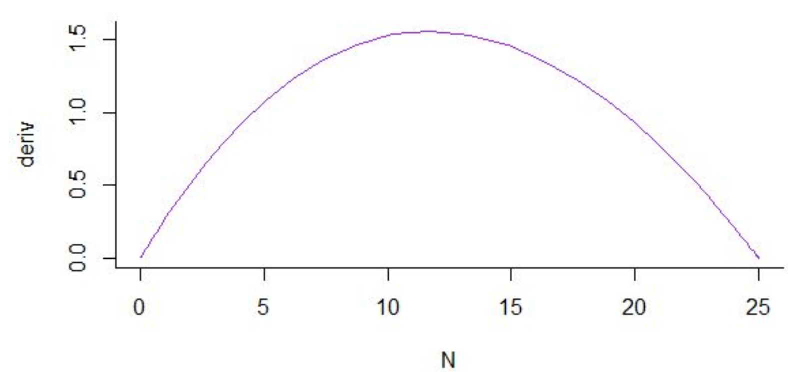
**Code for 'r' = 0.25, 'K' = 50:**

```
## theta log growth function
log.growth <- function(t, y, p) {
  N <- y[1]
  with(as.list(p), {
    dN.dt <- r * N * (1 - (N / K))
    return(list(dN.dt))
  })
}

## parameters
p.2 <- c('r' = 0.25, 'K' = 50)
y0 <- c('N' = runif(1, min = 0.01, max = 0.1))
t <- 1:100

## simulation
sim.2 <- ode(y = y0, times = t, func = log.growth, parms = p.2, method = 'lsoda')
sim.2 <- as.data.frame(sim.2)

## plot
plot(N ~ time, data = sim.2, type = 'l', lwd = 2, bty = 'l', col = 'blue')

## compute derivatives
sim.2$deriv <- c(diff(sim.2$N), NA)

## plot vs pop abundance
plot(deriv ~ N, data = sim.2, type = 'l', col = 'blue', bty = 'l')

## max value
max(sim.2$deriv, na.rm = TRUE)
##Output: 3.117201
which(sim.2$deriv == max(sim.2$deriv, na.rm = TRUE))
## Output: 27

## value of N in sim with the same index as the biggest deriv value
sim.2$N[which(sim.2$deriv == max(sim.2$deriv, na.rm = TRUE))]
## Output: 24.30376
```

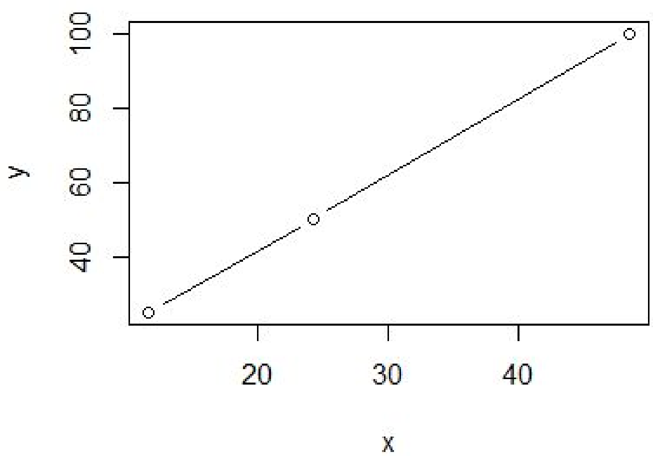



**Code for 'r' = 0.25, 'K' = 25:**

```
## theta log growth function
log.growth <- function(t, y, p) {
  N <- y[1]
  with(as.list(p), {
    dN.dt <- r * N * (1 - (N / K))
    return(list(dN.dt))
  })
}

## parameters
p.3 <- c('r' = 0.25, 'K' = 25)
y0 <- c('N' = runif(1, min = 0.01, max = 0.1))
t <- 1:100

## simulation
```



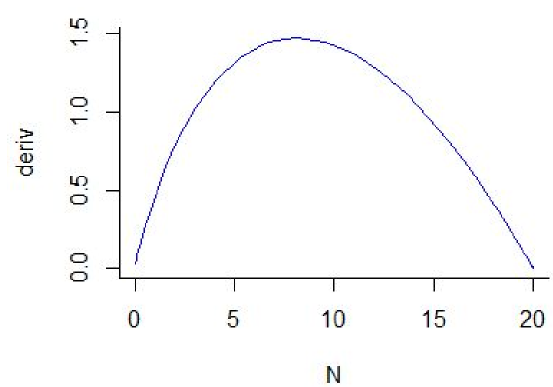
```
sim.3 <- ode(y = y0, times = t, func = log.growth, parms = p.3, method = 'lsoda')
sim.3 <- as.data.frame(sim.3)

## plot
plot(N ~ time, data = sim.3, type = 'l', lwd = 2, bty = 'l', col = 'purple')

## compute derivatives
sim.3$deriv <- diff(sim.3$N)
sim.3$deriv <- c(diff(sim.3$N), NA)

## plot vs pop abundance
plot(deriv ~ N, data = sim.3, type = 'l', col = 'purple',
bty = 'l')

## max value
max(sim.3$deriv, na.rm = TRUE)
## Output: 1.560464
which(sim.3$deriv == max(sim.3$deriv, na.rm =
TRUE))
##Output: 29

## value of N in sim with the same index as the
biggest deriv value
sim.3$N[which(sim.3$deriv == max(sim.3$deriv,
na.rm = TRUE))]
## Output: 11.74914


## plot these values against their corresponding K
parameter
x <- c(48.43107, 24.30376, 11.74914) ## N values
y <- c(100, 50, 25) ## K values
plot(x, y, type = 'b')
```




3.

A) Theta value: 0.5

```
log.growth <- function(t, y, p) {
  N <- y[1]
  with(as.list(p), {
    dN.dt <- r * N * (1 - (N / K)^theta)
    return(list(dN.dt))
  })
}

p <- c('r' = 0.5, 'K' = 20, 'theta' = 0.5)
y0 <- c('N' = 0.05)
t <- 1:100

sim <- ode(y = y0, times = t, func = log.growth, parms = p, method =
'lsoda')
sim <- as.data.frame(sim)
plot(N ~ time, data = sim, type = 'l', lwd = 2, bty = 'l', col = 'blue')

sim$deriv <- c(diff(sim$N), NA)
plot(deriv ~ N, data = sim, type = 'l', col = 'blue', bty = 'l')

max(sim$deriv, na.rm = TRUE)
##Output: 1.478818

which(sim$deriv == max(sim$deriv, na.rm = TRUE))
```

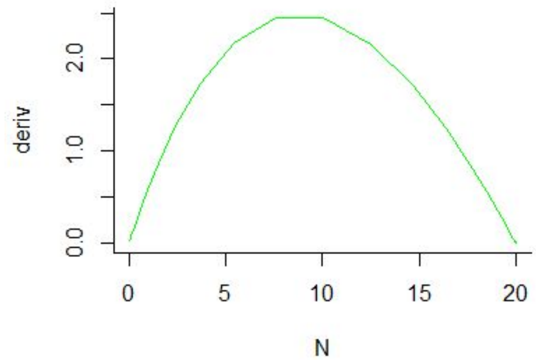


*##Output:15*

```
sim$N[which(sim$deriv == max(sim$deriv, na.rm = TRUE))]
```
*##Output: 8.075343*

## B) Theta value: 1

```
log.growth <- function(t, y, p) {
  N <- y[1]
  with(as.list(p), {
    dN.dt <- r * N * (1 - (N / K)^theta)
    return(list(dN.dt))
  })
}

p <- c('r' = 0.5, 'K' = 20, 'theta' = 1)
y0 <- c('N' = 0.05)
t <- 1:100

sim <- ode(y = y0, times = t, func = log.growth, parms = p, method =
'lsoda')
sim <- as.data.frame(sim)
plot(N ~ time, data = sim, type = 'l', lwd = 2, bty = 'l', col = 'green')

sim$deriv <- c(diff(sim$N), NA)
plot(deriv ~ N, data = sim, type = 'l', col = 'green', bty = 'l')
```



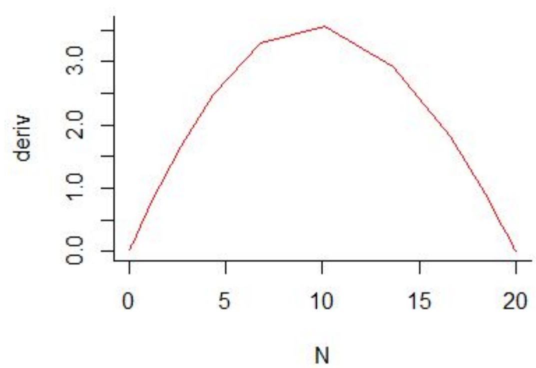

```
max(sim$deriv, na.rm = TRUE)
```
*##Output: 2.452442*

```
which(sim$deriv == max(sim$deriv, na.rm = TRUE))
```
*##Output: 12*

```
sim$N[which(sim$deriv == max(sim$deriv, na.rm = TRUE))]
```
*##Output: 7.602863*

## C) Theta value: 1.8

```
log.growth <- function(t, y, p) {
  N <- y[1]
  with(as.list(p), {
    dN.dt <- r * N * (1 - (N / K)^theta)
    return(list(dN.dt))
  })
}

p <- c('r' = 0.5, 'K' = 20, 'theta' = 1.8)
y0 <- c('N' = 0.05)
t <- 1:100

sim <- ode(y = y0, times = t, func = log.growth, parms = p, method =
'lsoda')
sim <- as.data.frame(sim)
plot(N ~ time, data = sim, type = 'l', lwd = 2, bty = 'l', col = 'red')

sim$deriv <- c(diff(sim$N), NA)
plot(deriv ~ N, data = sim, type = 'l', col = 'red', bty = 'l')
```




```
max(sim$deriv, na.rm = TRUE)
```
*##Output: 3.568666*

```
which(sim$deriv == max(sim$deriv, na.rm = TRUE))
```

*##Output: 12*

```
sim$N[which(sim$deriv == max(sim$deriv, na.rm = TRUE))]
```

*##Output: 10.09747*

According to the calculations, species C maintained the highest population abundance in the fishery (10.09747).