# Projecting biomass dynamics to find biological reference points

Analytical solutions exist to determine biological reference points like , , and for biomass dynamics models like the Graham Schaefer, Fox, and the Pella Tomlinson. However there are instances where analytical solutions might not be available because the underlying biomass dynamics model was modified to account for more biological realism. Additionally, the analytical solutions for biological reference points assume continuous biomass dynamics and therefore if a difference equation is used to project biomass dynamics, biological reference points may be biased. This box demonstrates how to find the biological reference points for a biomass dynamics model by simulation of a biomass dynamics model implemented as an ordinary differential equation or as a difference equation.

This box uses the Fox biomass dynamics model to project biomass dynamics over 100 years for varying fishing mortality rates (). The time horizon of 100 years was sufficient for this example for the biomass dynamics to reach equilibrium (i.e., change in biomass from one time point to the next is ~0). There are 2 examples, one for the Fox model implemented as an ODE and another with the Fox model implemented as a difference model.

## Libraries needed for analysis

The libraries listed in the code chunk below are required to plot the biomass dynamics and numerically integrate the biomass dynamics models.

library(lattice)  
library(deSolve)

## Evaluated reference points for a continuous biomass dynamics model

The function below sets up the Fox model of biomass dynamics as an ODE.

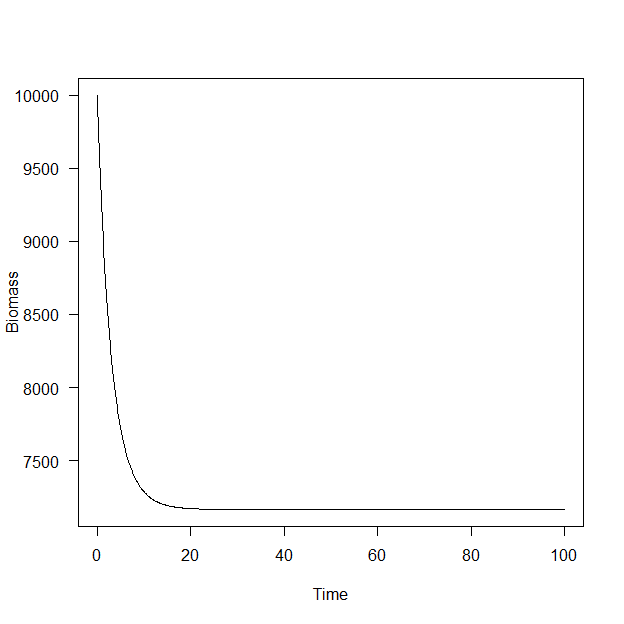
ode\_fox<-function(t,x,parms)  
 {  
 # set the state variable   
 B<-x   
   
 # set the parameters  
 r<-parms[1]  
 Bmax<-parms[2]  
 F<-parms[3]  
   
 # the Fox model of biomass dynamics as a   
 # ordinary differential equation  
 yield<- F\*B  
 dB<- r\*B\*(log(Bmax/B)) - F\*B  
 return(list(c(dB),Y=as.vector(yield)))# as.vector drops Y.F name baggage  
 }

The code chunk below sets the parameters of the model, initial biomass, and the then numerically integrates the model using the lsoda numerical integration routine. The projection biomass dynamics are saved as a data.frame.

parameters<-c(r=0.3,Bmax=10000,F=0.1)  
initial\_biomass<- c(B=10000)  
solution<- ode(  
 y=initial\_biomass,   
 times=seq(0,100,by=0.1),   
 func=ode\_fox,   
 parms=parameters,   
 method="lsoda")  
solution<-as.data.frame(solution)

The code chunk below plots the biomass dynamics. Looking at the biomass dynamics approaching year 100 there is no visible change in biomass, suggesting equilibrium has been achieved.

plot(B~time,solution,ylab="Biomass",  
 xlab="Time",las=1,main="",type="l")



The code chunk below returns the last 6 rows of the solution and the yield or biomass values do not change appreciably confirming that the biomass dynamics are in what is called approximate equilibrium (i.e., changes in biomass dynamics are too small to be consequential).

tail(solution)

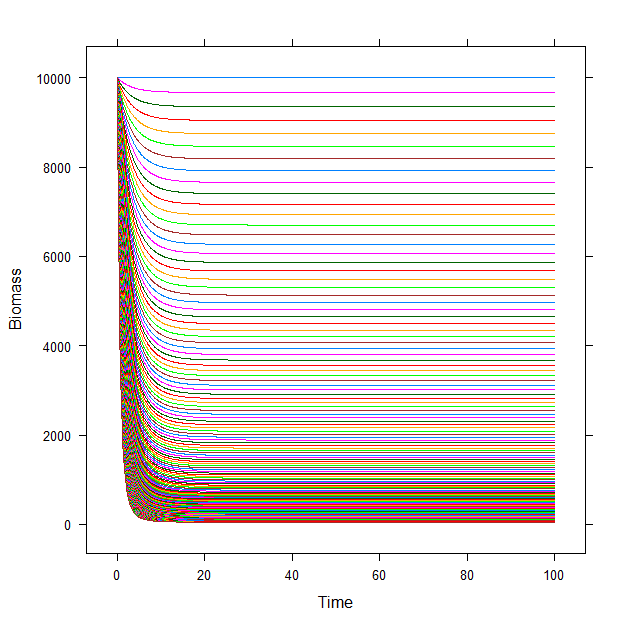
## time B Y  
## 996 99.5 7165.313 716.5313  
## 997 99.6 7165.313 716.5313  
## 998 99.7 7165.313 716.5313  
## 999 99.8 7165.313 716.5313  
## 1000 99.9 7165.313 716.5313  
## 1001 100.0 7165.313 716.5313

To evaluate biological reference points the biomass and yield at approximate equilibrium need to be identified for varying from 0 to a value that that exceeds the level that overfishing occurs so that the maximum sustained yield can be identified. This is don be looping over values and projecting the biomass dynamics out for a sufficiently long time, 100 years in this example. The code chunk below performs the loop and numerical integration. The results are collected in a data.frame using the rbind function for each value of evaluated.

F\_eval<-seq(0,1.6,by=0.01)  
out<-data.frame()  
for(i in 1:length(F\_eval))  
 {  
 parameters<-c(r=0.3,Bmax=10000,F=F\_eval[i])  
 initial\_biomass<- c(B=10000)  
 solution<- ode(  
 y=initial\_biomass,   
 times=seq(0,100,by=0.1),   
 func=ode\_fox,   
 parms=parameters,   
 method="rk4")  
 solution<-as.data.frame(solution)  
 solution$F<- F\_eval[i]  
 out<-rbind(out,solution)  
 }

The code chunk below plots the biomass dynamics for each value evaluated for visual evaluation to ensure approximate equlibrium was reached for all values.

xyplot(B~time,out,group=F,type="l",xlab="Time",  
 ylab="Biomass")



A quick look at the last 6 rows of the dataset we can see that the last time step evaluated is 100. We can use the subset function to extract the last value of biomass and yield for each value evaluated.

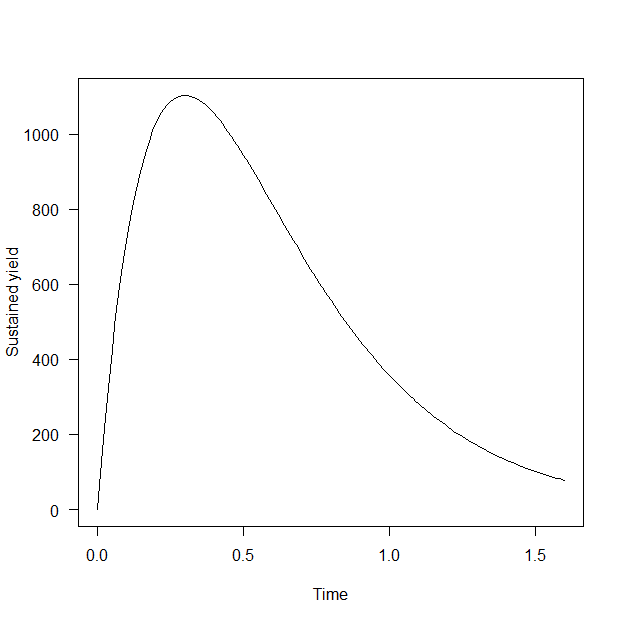
tail(out)

## time B Y F  
## 161156 99.5 48.2795 77.2472 1.6  
## 161157 99.6 48.2795 77.2472 1.6  
## 161158 99.7 48.2795 77.2472 1.6  
## 161159 99.8 48.2795 77.2472 1.6  
## 161160 99.9 48.2795 77.2472 1.6  
## 161161 100.0 48.2795 77.2472 1.6

equilibrium\_values<- subset(out, time==100)

Plotting yield versus illustrates the sustained or equilibrium yield curve. The top of the curve is the maximum sustained yield ().

# plot equilibrium yield values  
plot(Y~F,equilibrium\_values,ylab="Sustained yield",  
 xlab="Time",las=1,main="",  
 type='l')



## Biological reference points

Biological reference points can be calculated from the equilibrium yield curve. Specifically, the biomass and values associated with the can be identified by using the which.max function to return the row of a data.frame that has the maximum of a vector.

# get biomass and F that maximize equilibrium yield  
bio\_ref\_points<-equilibrium\_values[which.max(equilibrium\_values$Y),]

The value contained in bio\_ref\_points are and .

bio\_ref\_points

## time B Y F  
## 31031 100 3678.794 1103.638 0.3

## Finding

The numerical approach can also be used to identify the more conservative . The code chunk below calculates the slope over the equilibrium yield curve. The slope at the origin is then multiplied by 0.1 and the tangent line on the curve identified.

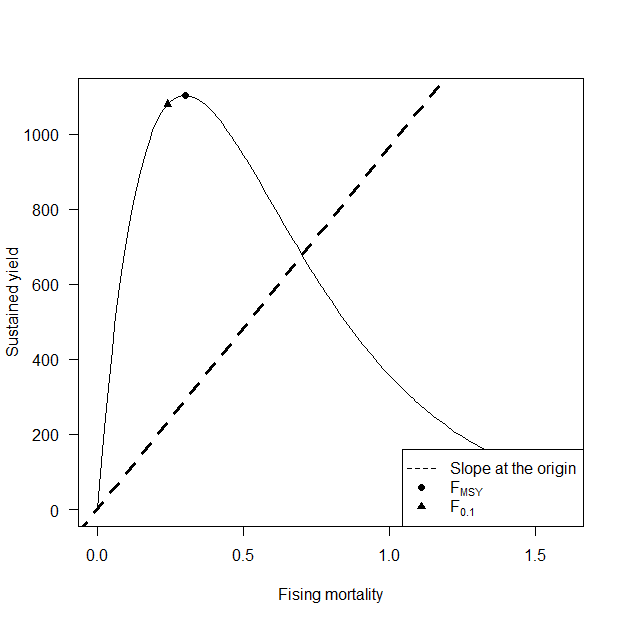
equilibrium\_values$slopes<- c(0,diff(equilibrium\_values$Y)/0.01)  
slope10<-equilibrium\_values$slopes[2]\*0.1

The approxfun function is useful here to interpolate values. The code chunk below sets up an interpolation function that returns an and yield value associated with a slope. The code chunk below returns the and yield associated with 10% of the yield slope at the origin.

find\_F<- approxfun(equilibrium\_values$slopes,equilibrium\_values$F)  
f\_01<-find\_F(slope10)  
find\_Y<- approxfun(equilibrium\_values$slopes,equilibrium\_values$Y)  
Y\_01<-find\_Y(slope10)

The plot below shows the 10% of the slope at the origin, the value that maximizes sustained yield () and where the slope of the tangent line of sustained yield curve is 10% of the slope of a tangent at the origin.

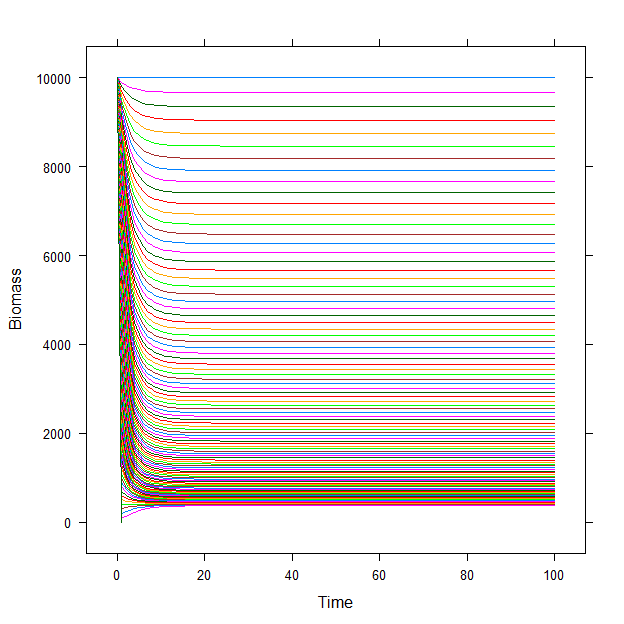
plot(Y~F,equilibrium\_values,ylab="Sustained yield",  
 xlab="Fising mortality",las=1,main="",  
 type="l")  
abline(0,slope10,lwd=3,lty=2)  
points(Y~F,bio\_ref\_points,pch=19)  
points(f\_01,Y\_01,pch=17)  
legend("bottomright", lty=c(2,NA,NA),pch=c(NA,19,17),  
 legend=c("Slope at the origin",expression("F"[MSY]), expression("F"[0.1])))



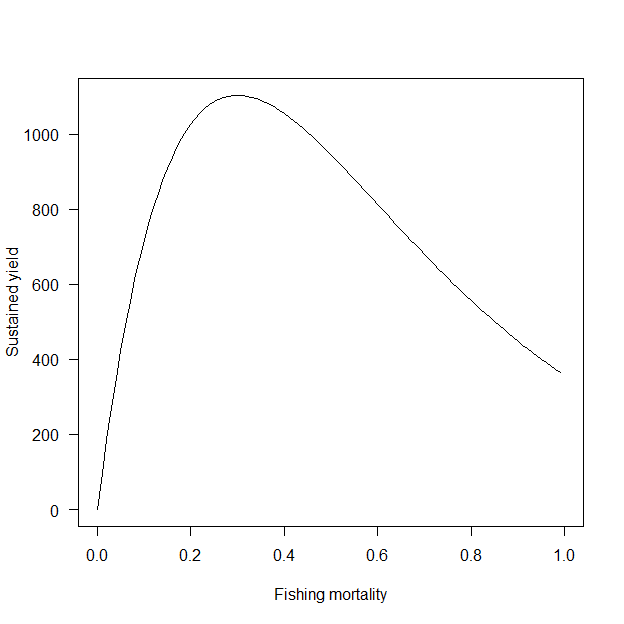
## Evaluated reference points for a discrete biomass dynamics model

The code chunk below replicates the analysis done above but for the Fox model implemented as a difference model assuming a discrete time step of 1 year.

diff\_fox<-function(t,x,parms)  
 {  
 # set the state variable   
 B<-x   
   
 # set the parameters  
 r<-parms[1]  
 Bmax<-parms[2]  
 F<-parms[3]  
   
 # the Fox model of biomass dynamics as a   
 # ordinary differential equation  
 yield<- F\*B  
 B<-B+r\*B\*(log(Bmax/B)) - F\*B  
 return(list(c(B),Y=as.vector(yield)))# as.vector drops Y.F name baggage  
 }  
# evaluate sustained yield  
F\_eval<-seq(0,1,by=0.01)  
out<-data.frame()  
for(i in 1:length(F\_eval))  
 {  
 parameters<-c(r=0.3,Bmax=10000,F=F\_eval[i])  
 initial\_biomass<- c(B=10000)  
 solution<- ode(  
 y=initial\_biomass,   
 times=seq(0,100,by=1),   
 func=diff\_fox,   
 parms=parameters,   
 method="iteration")  
 solution<-as.data.frame(solution)  
 solution$F<- F\_eval[i]  
 out<-rbind(out,solution)  
 }  
# look at sustained yield by F  
xyplot(B~time,out,group=F,type="l",xlab="Time",  
 ylab="Biomass")



# subset values at year 100  
equilibrium\_values<- subset(out, time>99.9999)  
# plot equilibrium values  
plot(Y~F,equilibrium\_values,ylab="Sustained yield",  
 xlab="Fishing mortality",las=1,main="",  
 type="l")



# subset biological reference points  
bio\_ref\_points<-equilibrium\_values[which.max(equilibrium\_values$Y),]  
bio\_ref\_points

## time B Y F  
## 3131 100 3678.794 1103.638 0.3

# find F\_0.1  
equilibrium\_values$slopes<- c(0,diff(equilibrium\_values$Y)/0.01)  
slope10<-equilibrium\_values$slopes[2]\*0.1  
find\_F<- approxfun(equilibrium\_values$slopes,equilibrium\_values$F)  
f\_01<-find\_F(slope10)  
find\_Y<- approxfun(equilibrium\_values$slopes,equilibrium\_values$Y)  
Y\_01<-find\_Y(slope10)  
# illustrate F\_MSY and F\_0.1  
plot(Y~F,equilibrium\_values,ylab="Sustained yield",  
 xlab="Fising mortality",las=1,main="",  
 type="l")  
points(Y~F,bio\_ref\_points,pch=19)  
points(f\_01,Y\_01,pch=17)  
legend("bottomright", pch=c(19,17),  
 legend=c(expression("F"[MSY]), expression("F"[0.1])))

