```
# Non-linear Model Example  - Bruchid Data and Fitting Algorithm - R Code

#read in bruchid data and plot time series

quartz()
par(mfrow=c(1,1))
bruchids<-scan("bruchids.txt")
plot(1:length(bruchids),bruchids,xlab="Time (Weeks)", ylab="Abundance
(Alive  Counts)",type="b")
bruchids<-log(bruchids)

#define Poisson likelihood function for correlated errors with unknown
parameter vector p
#define the model as a non-linear density-dependent model

fn=function(p){
mod=array(0,dim=c(77))
mod[1]=(p[1]^(1/p[3])-1)/p[2]
tmp=0
for(i in 2:77){
mod[i]=p[1]*mod[i-1]/(1+p[2]*mod[i-1])^p[3]
likl=(lfactorial(bruchids[i])+mod[i]-log(mod[i])*bruchids[i])
tmp=tmp+likl}
tmp+(lfactorial(bruchids[1])+mod[1]-log(mod[1])*bruchids[1])
}

#define the initial parameter set
p<-c(2.75,0.03,1.0)

#use appropriate optimization routine (in R - default is Nelder-Mead)
out=optim(p,fn)

#displays parameter estimates
out$par

#produces one-step ahead predictions from ML parameter estimates

x=array(0,dim=c(77))
x[1]=mean(bruchids)
for(i in 2:77)
        x[i]<-out$par[1]*bruchids[i-1]/(1+out$par[2]*bruchids[i-1])^out$par
[3]

        plot(seq(1,77,length=10),seq
(1,5,length=10),type="n",bty="l",xlab="Time(Weeks)",ylab="log(Abundance)")
        lines(c(1:77),bruchids,lwd=2,col="purple")
        points(c(1:77),x,pch=19,cex=1.0)

quartz()
par(mfrow=c(1,2))
plot(c(1:77),bruchids-x,type="p",bty="l",xlab="Time",ylab="Raw
Residuals",pch=19,cex=1.0)
plot(x,bruchids-x,type="p",bty="l",xlab="Fitted Values",ylab="Raw
Residuals",pch=19,cex=1.0)
```