Population Dynamics of Macrophytes in a Shallow Water Lake

Jasmine Fowler-Morrow

2022-05-25

Introduction

The abundance of aquatic vegetation (macrophytes) in shallow lakes play a key role in reducing turbidity and providing a food source for inhabitants. However, disturbing or removing macrophytes can cause rapid and irreversible changes to conditions (Scheffer, 2004). Such observations have led researchers to conclude that shallow lakes commonly have two stable states; clear and turbid (Scheffer, 2004).The state of the ecosystem will change after a critical value, or tipping point, is reached.

Previous research found macrophyte population size to primarily determine whether or not a state shift occurred (Scheffer, 1993). Dakos et al (2019) further argued that trait variation and evolution are important for understanding tipping point dynamics. Dakos et al (2019) found that phenotypic variation can make macrophytes more resilient to collapse. However, this model was studied analytically, with a fixed trait score for the entire population which remained constant through time. This is a limited approach, since individuals who are fitter should be more likely to survive and reproduce.

Methods

I created an individual-based stochastic model describing macrophyte population dynamics in response to turbidity. The model is adapted from an ordinary differential equation system described by Dakos et al (2019). The model tracks a population of macrophyte individuals and their trait value (z) through time. The simulation began with an initial population (n_0) of macrophyte individuals, each with a trait value (z). Initial trait values were generated with a beta distribution (shape = z_1, scale =z_2). Each time period, an individual has a fixed probability of dying (μ). Given an individual does not die, they have a probability of reproducing:

$$P(reproducing) = r_M M \left(1 - \frac{M}{K}\left(\frac{h_T^4 + T^4}{h_t^4}\right)\right)$$

This probability was used to closely match the analytic model, since highest reproduction probability would occur when growth rate was highest. Reproduction depended on a number of fixed parameters (Table 1), the number of macrophyte individuals in the previous time period (M), sensitivity to turbidity (h_T) and turbidity in the current time period (T). Turbidity was updated each time using a difference equation:

$$T_{t+1} = T_t + r_T T_t \left(1 - \frac{T_t}{T_0 \frac{h_M}{h_M + \frac{M}{K}}}\right)$$

Sensitivity to turbidity was determined by individual trait value (h_T) in a relationship described by Dakos et al (2019):

$$h_T = e^{cz}$$

With parameters also fixed (table 1). Consequently, higher turbidity and lower trait values decreased an individuals chance of reproducing. It was assumed that all reproduction was asexual, and offspring received the same trait value as their parent with some random genetic mutation (σ).

Table 1: Parameter values/ranges used in the model.*values selected by Dakos et al (2019)

Parameter	Value/Range
Carrying capacity (K)	50
Growth rates (r_M, r_T)	0.1*
Turbidity half-saturation parameter h_M	0.2*
Trait value (z)	$[-2, 2]^*$
Death rate (μ)	0.05
Initial turbidity (initurb)	$0.1, 0.3, 0.6, 1$
Background turbidity (T_0)	$[2, 8]^*$
Genetic mutation rate (σ)	0.01
turbidity response (c)	0.5

All analysis was conducted in RStudio. To assess macrophyte success, average population size was recorded through time. Turbidity level and average trait values were also recorded. I used these outputs to assess how background turbidity, trait variation and initial macrophytes and turbidity influenced results.

Code

```r
library(tidyverse)
library(reshape2)
#install.packages("patchwork")
library(patchwork)
```

```r
############## Simulation function #####################
#function to simulate macrophyte individuals and their trait values over time in response to turbidity
#Inputs:
#inipop = dataframe of initial macrophyte population
#initurb = number describing the starting turbidity levels
#p = list of parameters
#returns: macrophyte number, turbidity level and trait (z) mean and sd

macrophyte_growth <- function(inipop, initurb, tmax, p){
  ###set up data###
  Turb <- rep(0, tmax) #store turbidity over simulation
  Turb[1] <- initurb #initial Turbidity level

  #macrophyte population
  M <- rep(0, tmax) #store population size over simulation
  M[1] <- length(inipop$ID) #initial population size
  mtt <- as.vector(rep(NA, tmax), "list") #macrophytes through time
  mtt[[1]] <- inipop

  #trait value summaries
  Z_means <- rep(0, tmax)
  Z_means[1] <- mean(inipop$z)
  Z_sd <- rep(0, tmax)
  Z_sd[1] <- sd(inipop$z)
```

```
for (i in 1:(tmax-1)){
  #create empty data frame with empty rows for reproduction
  mtt[[i+1]] <- data.frame(ID = rep(NA,p$K +20),
                     z = NA,
                     alive = NA)
  mtt[[i+1]][1:nrow(mtt[[i]]),] <- mtt[[i]] #copy population over to next time step

  #update turbidity level
  Turb[i+1] <- Turb[i] +
    p$rt*Turb[i]*(1-(Turb[i]/(p$T0*(p$hm/(p$hm +(M[i]/p$K))))))

  #determine the outcome of each individual in this time-step
  for (j in 1:length(mtt[[i+1]]$ID[!is.na(mtt[[i + 1]]$ID)])){
    #each individual has probability of death
    mtt[[i+1]]$alive[j] <- rbernoulli(1,1-p$mu)

    if (mtt[[i+1]]$alive[j]){
      #if alive there is a chance of asexual reproduction
      #ht value for that individual
      ht <- exp(0.5*mtt[[i+1]]$z[j])
      reproduce <- rbernoulli(1,
                              p$rm*M[i]*(1-(M[i]/p$K)*((ht^4+Turb[i+1])/ht^4)))

      if (reproduce) { #create new individual
        emptySlot <- which(is.na(mtt[[i + 1]]$ID))[1] # next empty row in dataframe
        mtt[[i+1]]$ID[emptySlot] <- last(mtt[[i+1]]$ID[!is.na(mtt[[i + 1]]$ID)]) + 1
        mtt[[i+1]]$z[emptySlot] <- mtt[[i+1]]$z[j] + rnorm(1,sd = p$sigma)
        mtt[[i+1]]$alive[emptySlot] <- TRUE

      }
    }
  }
  #remove rows with na's
  mtt[[i + 1]] <- mtt[[i + 1]][!is.na(mtt[[i + 1]]$ID),]
  #remove dead individuals
  mtt[[i + 1]] <- mtt[[i + 1]][mtt[[i + 1]]$alive,]
  #new macrophyte population size
  M[i+1] <- length(mtt[[i+1]]$ID)

  # check whether population is extinct
  if (M[i+1]==0){
    return(list(M = M, Turb = Turb, Z_means = Z_means, Z_sd = Z_sd))
    stop("Macrophyte population collapse")
  }

  #otherwise, update population information
  else{
    Z_means[i+1] <- mean(mtt[[i+1]]$z) #new population trait val mean
    Z_sd[i+1] <- sd(mtt[[i+1]]$z) #new population trait sd
  }

}
return(list(M = M, Turb = Turb, Z_means = Z_means, Z_sd = Z_sd))
```

```
}
```

```
######## Running function ###########
#this function runs the macrophyte population model numerous times
#it allows parameters and initial values to be specified
# it requires number of simulations (n_sims)
#returns 4 matrices corresponding to macrophyte population, turbidity level, mean trait value and trait
runMacrophyte <- function(n_sims, tmax = 100, rm = 0.1, rt = 0.1,
                          hm = 0.2, T0 = 3, z1 = 2, z2 = 2,
                          K = 50, mu = 0.05, sigma = 0.001, n0 = 10,initurb = 0.5){
  #create parameter list
  p <- list() #parameter list
  p$rm <- rm #macrophyte growth rate
  p$rt <- rt #turbidity growth rate
  p$hm <- hm #macrophyte half saturation
  p$T0 <- T0 #background turbidity
  p$K <- K #macrophyte carrying capacity
  p$mu <- mu #probability of death
  p$sigma <- sigma

  #create data frame for initial population
  inipop <- data.frame(ID = 1:n0,
                       z = 2*(-0.5+rbeta(n0, z1, z2)), #generate trait values
                       alive = TRUE)

  ##matrices to store simulation results
  mac_pop <- matrix(nrow = n_sims,ncol = tmax)
  turb <- matrix(nrow = n_sims,ncol = tmax)
  Z_means <- matrix(nrow = n_sims,ncol = tmax)
  Z_sd <- matrix(nrow = n_sims,ncol = tmax)

  ##run simulations and store relevant results
  for (i in 1:n_sims){
    results <- macrophyte_growth(inipop, initurb, tmax, p)
    mac_pop[i,] <- results$M
    turb[i,] <- results$Turb
    Z_means[i,] <- results$Z_means
    Z_sd[i,] <- results$Z_sd
  }
  return(list(mac_pop = mac_pop,turb = turb, Z_means = Z_means, Z_sd = Z_sd))
}
```

```
#### Parameter screen - results with varying initial turbidity ######
#this function summarises the simulation results from different initial turbidity values
#returns: a plot of macrophyte numbers, turbidity and trait mean through time for each initial turbidit
vary_initial_turbidity <- function(){

  #initial turbidity
  turb <- c(0.1,0.3, 0.6, 1)
  #data frames for storing results
  macrophytes <- data.frame(NULL)
  turbidity <- data.frame(NULL)
  trait_val <- data.frame(NULL)
  for (t in 1:length(turb)){
    #simulate for turbidity value t
    result <- runMacrophyte(30, initurb = turb[t])

    ### MACROPHYTES ###
    df <- melt(result$mac_pop, varnames = c("Sim","Time"), value.name = "M") %>%
      group_by(Time) %>%
      summarise(mean_M = mean(M), se = sd(M)/sqrt(length(M))) %>%
      add_column(initurb = turb[t])
    macrophytes <- rbind(macrophytes, df) #include current simulation in macrophyte data

    ### TURBIDITY ###
    df2 <- melt(result$turb, varnames = c("Sim","Time"), value.name = "Tb") %>%
      group_by(Time) %>%
      summarise(mean_T = mean(Tb), se = sd(Tb)/length(Tb))%>%
      add_column(initurb = turb[t])
    turbidity <- rbind(turbidity,df2) #include current simulation in turbidity data

    ### TRAIT MEAN ###
    df3 <- melt(result$Z_means, varnames = c("Sim","Time"), value.name = "zbar")%>%
      group_by(Time) %>%
      summarise(mean_z = mean(zbar), se = sd(zbar)/length(zbar))%>%
      add_column(initurb = turb[t])
    trait_val <- rbind(trait_val,df3) #include current simulation in mean trait value data
}
### plot macrophyte numbers through time ###
  p1 <- ggplot() +
    geom_line(data = macrophytes, aes(x = Time, y = mean_M, group = initurb,
                                      colour = as.factor(initurb)), lwd = 1) +
    geom_ribbon(data = macrophytes, aes(x = Time, ymin = mean_M-se, ymax = mean_M+se,
                                        group = initurb,fill = as.character(initurb)),
                alpha = 0.3, show.legend = FALSE)+
    theme_minimal() + labs(x = "Time step", y = "Macrophyte individuals", colour = "Initial turbidity")+
      theme(legend.position = "bottom")
  ### plot turbditiy levels through time ###
  p2 <- ggplot() +
    geom_line(data = turbidity, aes(x = Time, y = mean_T, group = initurb,
                                    colour = as.factor(initurb)), lwd = 1) +
    geom_ribbon(data = turbidity, aes(x = Time, ymin = mean_T-se, ymax = mean_T+se,
                                        group = initurb,fill = as.character(initurb)),
                alpha = 0.3, show.legend = FALSE)+
    theme_minimal()+ labs(x = "Time step", y = "Turbidity level", colour = "Initial turbidity")+
      theme(legend.position = "bottom")
  ### plot trait means (z) through time ###
  p3 <- ggplot() +
    geom_line(data = trait_val, aes(x = Time, y = mean_z, group = initurb,
                                    colour = as.factor(initurb)), lwd = 1) +
    geom_ribbon(data = trait_val, aes(x = Time, ymin = mean_z-se, ymax = mean_z+se,
                                        group = initurb,fill = as.character(initurb)),
                alpha = 0.3, show.legend = FALSE)+
    theme_minimal()+ labs(x = "Time step", y = "Mean trait value (z)", colour = "Initial turbidity")+
      theme(legend.position = "bottom")
  ### use patchwork to return ###
  p <- (p1 | p2 | p3) + plot_layout (guides = "collect") &
  theme(legend.position='bottom')
  return(p)

}

vary_initial_turbidity()
```

```
#### Parameter screen - results with varying initial macrophyte numbers ######
#this function summarises the simulation results from different initial macrophyte population sizes
#returns: a plot of macrophyte numbers, turbidity and trait mean through time for each initial populati
vary_initial_mac <- function(){

  #initial turbidity
  size <- c(5,10, 20, 40)
  macrophytes <- data.frame(NULL)
  turbidity <- data.frame(NULL)
  trait_val <- data.frame(NULL)

  for (n in 1:length(size)){
    #simulate for turbidity value t
    result <- runMacrophyte(30, n0 = size[n])

    ### MACROPHYTES ###
    df <- melt(result$mac_pop, varnames = c("Sim","Time"), value.name = "M") %>%
      group_by(Time) %>%
      summarise(mean_M = mean(M), se = sd(M)/sqrt(length(M))) %>%
      add_column(n0 = size[n])
    macrophytes <- rbind(macrophytes, df)

    ### TURBIDITY ###
    df2 <- melt(result$turb, varnames = c("Sim","Time"), value.name = "Tb") %>%
      group_by(Time) %>%
      summarise(mean_T = mean(Tb), se = sd(Tb)/length(Tb))%>%
      add_column(n0 = size[n])
    turbidity <- rbind(turbidity,df2)

    ### TRAIT MEAN ###
    df3 <- melt(result$Z_means, varnames = c("Sim","Time"), value.name = "zbar")%>%
      group_by(Time) %>%
      summarise(mean_z = mean(zbar), se = sd(zbar)/length(zbar))%>%
      add_column(n0 = size[n])
    trait_val <- rbind(trait_val,df3)
}
  ### plot macrophyte numbers through time ###
  p1 <- ggplot() +
    geom_line(data = macrophytes, aes(x = Time, y = mean_M, group = n0,
                                      colour = as.factor(n0)), lwd = 1) +
    geom_ribbon(data = macrophytes, aes(x = Time, ymin = mean_M-se, ymax = mean_M+se,
                                        group = n0,fill = as.factor(n0)),
                alpha = 0.3, show.legend = FALSE)+
    theme_minimal() + labs(x = "Time step", y = "Macrophyte individuals", colour = "Initial macrophyte
      theme(legend.position = "bottom")
  ### plot turbidity levels through time ###
  p2 <- ggplot() +
    geom_line(data = turbidity, aes(x = Time, y = mean_T, group = n0,
                                    colour = as.factor(n0)), lwd = 1) +
    geom_ribbon(data = turbidity, aes(x = Time, ymin = mean_T-se, ymax = mean_T+se,
                                        group = n0,fill = as.factor(n0)),
                alpha = 0.3, show.legend = FALSE)+
    theme_minimal()+ labs(x = "Time step", y = "Turbidity level", colour = "Initial macrophyte no.")+
      theme(legend.position = "bottom")
  ### plot mean trait vals through time ###
  p3 <- ggplot() +
    geom_line(data = trait_val, aes(x = Time, y = mean_z, group = n0,
                                    colour = as.factor(n0)), lwd = 1) +
    geom_ribbon(data = trait_val, aes(x = Time, ymin = mean_z-se, ymax = mean_z+se,
                                        group = n0,fill = as.factor(n0)),
                alpha = 0.3, show.legend = FALSE)+
    theme_minimal()+ labs(x = "Time step", y = "Mean trait value (z)", colour = "Initial macrophyte no.
      theme(legend.position = "bottom")

  ### use patchwork to return ###
  p <- (p1 | p2 | p3) + plot_layout (guides = "collect") &
  theme(legend.position='bottom')
  return(p)

}

vary_initial_mac()
```

```
#this function summarises the simulation results
#summary - "data" returns a list of data frames of population size, turbditiy and trait mean
#summary = "stats" returns mean final population size and turbidity and SE
Macrophyte_sim_summary <- function(result, summary = "stats"){
  ### MACROPHYTES ###
  df <- melt(result$mac_pop, varnames = c("Sim","Time"), value.name = "M")
  dfmean <- df %>%
    group_by(Time) %>%
    summarise(mean_M = mean(M), se = sd(M)/length(M))

  ### TURBIDITY ###
  df2 <- melt(result$turb, varnames = c("Sim","Time"), value.name = "Tb")
  df2mean <- df2 %>%
    group_by(Time) %>%
    summarise(mean_T = mean(Tb))

  ### TRAIT MEAN ###
  df3 <- melt(result$Z_means, varnames = c("Sim","Time"), value.name = "zbar")
  df3mean <- df3 %>%
    group_by(Time) %>%
    summarise(mean_z = mean(zbar))

  ### TRAIT SD ###
  df4 <- melt(result$Z_sd, varnames = c("Sim","Time"), value.name = "zsd")
  df4mean <- df4 %>%
    group_by(Time) %>%
    summarise(mean_zsd = mean(zsd))

   if (summary == "stats"){
    M_end <- dfmean[dfmean$Time==100,2]
    Turb_end <- df2mean[df2mean$Time==100,2]
    return(data.frame(M_end = M_end, Turb_end = Turb_end))
  }
  else if(summary == "data"){
    return(list(M = df, Tb = df2, z = df3))
  }

}
```

```
####### simulate final population size over range of background turbidity ######
####### WITH trait variation #######
M <- rep(0,7)
final_mac <- data.frame(T0 = NULL, final_M = NULL)
for (t in 2:8){
  #run simulation for given T0 value
  result <- runMacrophyte(50, initurb = 0.5, T0 = t, n0 = 10)

  #average final macrophyte population size
  M[t-1] <- Macrophyte_sim_summary(result, "stats")[[1]]

  #final macrophyte population size for each simulation
  Mac <- Macrophyte_sim_summary(result, summary = "data")[[1]]
  Mac <- cbind(T0 = rep(t,10), final_M = Mac[Mac$Time==100,"M"])
  #store final size with T0 level
  final_mac <- rbind(final_mac, Mac)

}

####### WITHOUT trait variation #######
M2 <- rep(0,7)
final_mac2 <- data.frame(T0 = NULL, final_M = NULL)
for (t in 2:8){
  #run simulation for given T0 value
  result <- runMacrophyte(50, initurb = 0.5, T0 = t, n0 = 10,
                          z1 = 100, z2 = 100)

  #average final macrophyte population size
  M2[t-1] <- Macrophyte_sim_summary(result, "stats")[[1]]

  #final macrophyte population size for each simulation
  Mac <- Macrophyte_sim_summary(result, summary = "data")[[1]]
  Mac <- cbind(T0 = rep(t,10), final_M = Mac[Mac$Time==100,"M"])
  #store final size with T0 level
  final_mac2 <- rbind(final_mac2, Mac)

}
```

```r
####### final population values expected from analytic solutions ########
analytic_M <- read.csv("analytic_M.csv")

## plot of background turbidity vs population size ##
ggplot()+geom_jitter(data = final_mac, aes(x = T0, y = final_M),
                     col = "#2c7bb6", alpha = 0.5, width = 0.1) +
  geom_line(aes(x = 2:8, y = M), col = "#2c7bb6", lwd = 1) +
  geom_point(aes(x = 2:8, y = M), col = "#2c7bb6") +
  labs(x = "Background Turbidity (T0)", y="Final macrophyte population size")+
  geom_line(aes(x = 2:8, y = M2), col = "#d7191c", lwd = 1) +
  geom_jitter(data = final_mac2, aes(x = T0, y = final_M),
                     col = "#d7191c", alpha = 0.5, width = 0.1)+
  geom_point(data = analytic_M, aes(x = T0, y = M), col = "#404040")+
  geom_line(data = analytic_M[-2,], aes(x = T0, y = M), col = "#404040",
            lwd = 0.7, lty = 2) +
  geom_line(data = analytic_M[-3,], aes(x = T0, y = M), col = "#404040",
            lwd = 0.7, lty = 2) +
  theme_minimal()
```

Results

Background turbidity level negatively impacted final macrophyte population size (figure 1). However, this decrease is not as pronounced as the analytic model suggested; population collapses were not observed in the extended model results. Furthermore, populations with higher initial trait variation were less affected by background turbidity (figure 1).

Initial turbidity had no evident effect on macrophyte population dynamics (figure 2). However, differences of in the final turbidity and population size vary based on their initial trait averages.

Initial population size did influence the ecosystem state; smaller initial populations reached lower final population sizes (figure 3). Turbidity also remained higher in smaller populations.

Conclusion

Macrophyte population dynamics change when stochastic and individual variation was considered. Larger trait variation in the initial macrophyte population increased their resilience to environmental conditions. Consequently, trait variation is extremely important for maintaining shallow lake ecosystems.

Initial values had mixed effects; initial turbidity did not influence macrophyte population trajectories, however smaller initial populations performed worse on average. This information is important for restoration efforts - reintroduction of macrophytes should be possible at various turbidity levels. Smaller initial population sizes performed worse on average, suggesting that …

Bibliography

Dakos, Matthews, B., Heny, A. P., Levine, J., Loeuille, N., Norberg, J., Nosil, P., Scheffer, M., & Meester, De, Luc. (2019). Ecosystem tipping points in an evolving world. *Nature Ecology & Evolution*, 3(3), 355–362.

Scheffer, DeAngelis, D. L., & Manly, B. J. F. (2004). *Ecology of Shallow Lakes* (Vol. 22). Springer Netherlands.

Scheffer, Hosper, S. ., Meijer, M.-L., Moss, B., & Jeppesen, E. (1993). Alternative equilibria in shallow lakes. *Trends in Ecology & Evolution*, 8(8), 275–279.

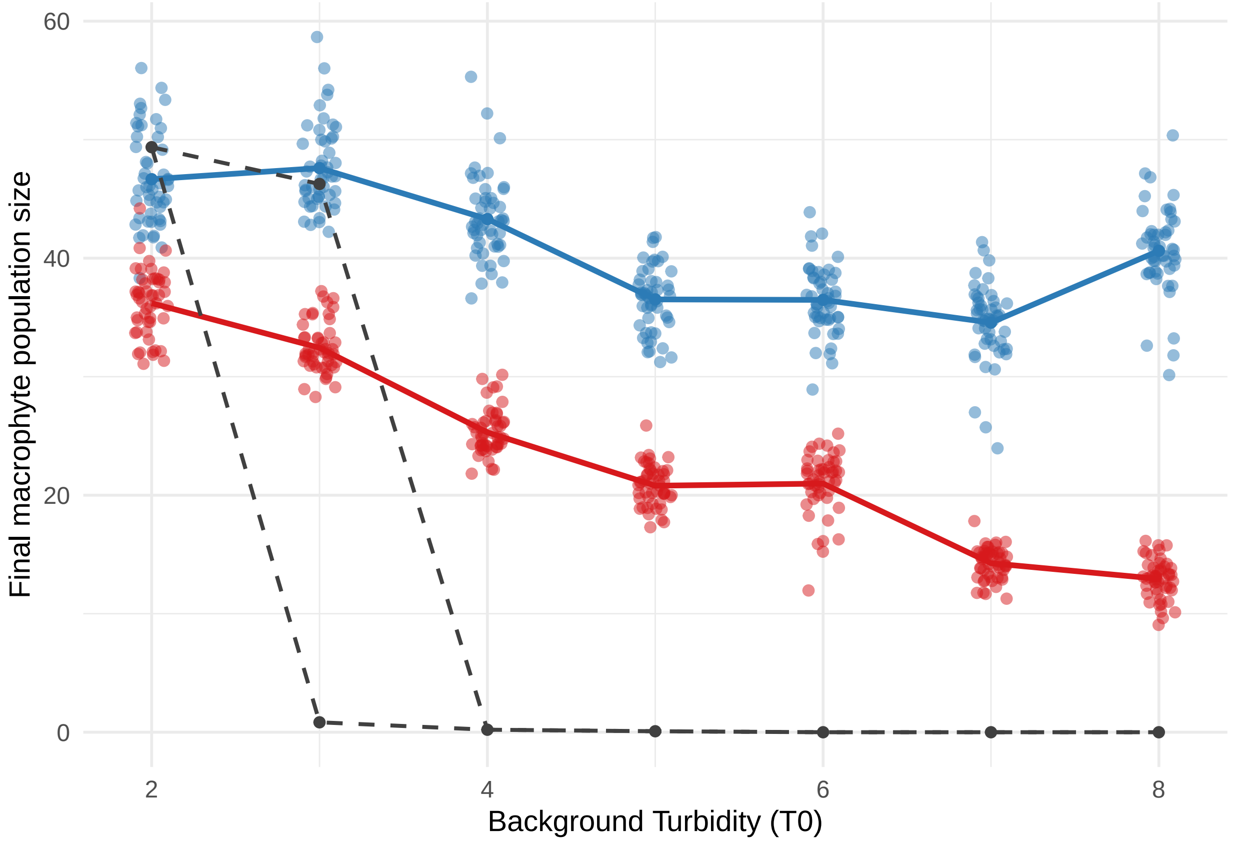

Figure 1: The effect of background turbidity levels on final macrophyte population size. Equilibria from the analytic model are pictured for each parameter value (black points). Simulation results are represented by transparent points for low trait variance (red, simulated with beta(100,100)) and high variance (blue, simulated with beta(2,2)). Averages for each group at each value are connected by solid lines (n=30).

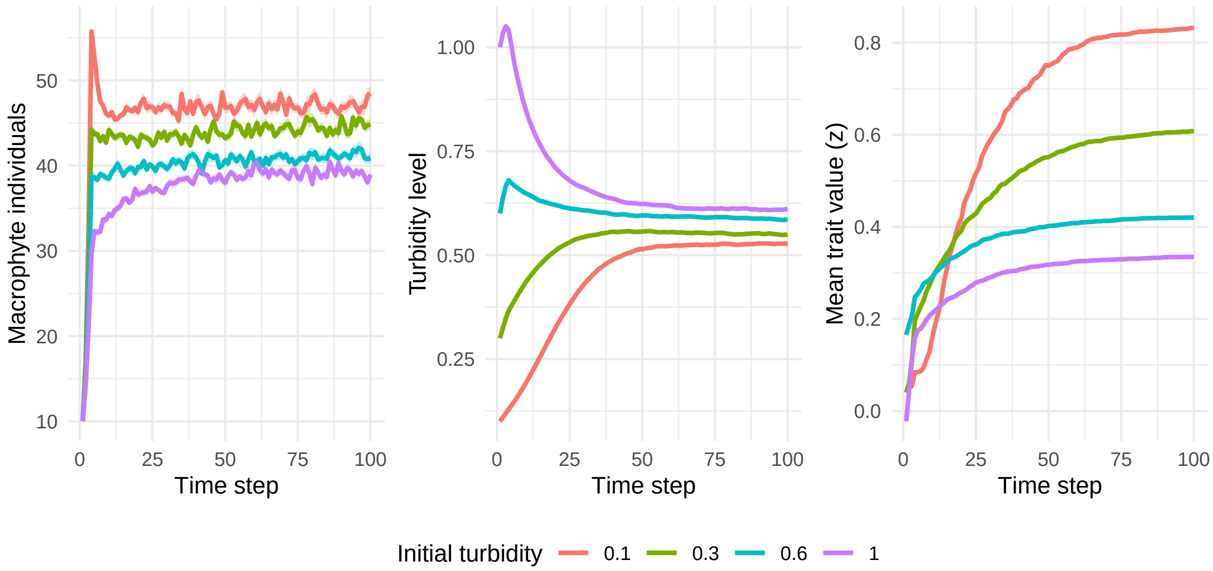

Figure 2: Macrophyte population size, turbidity level and trait value over time for various initial turbidity levels, indicated by colour. Solid lines indicated an average value (n = 30) at each time.

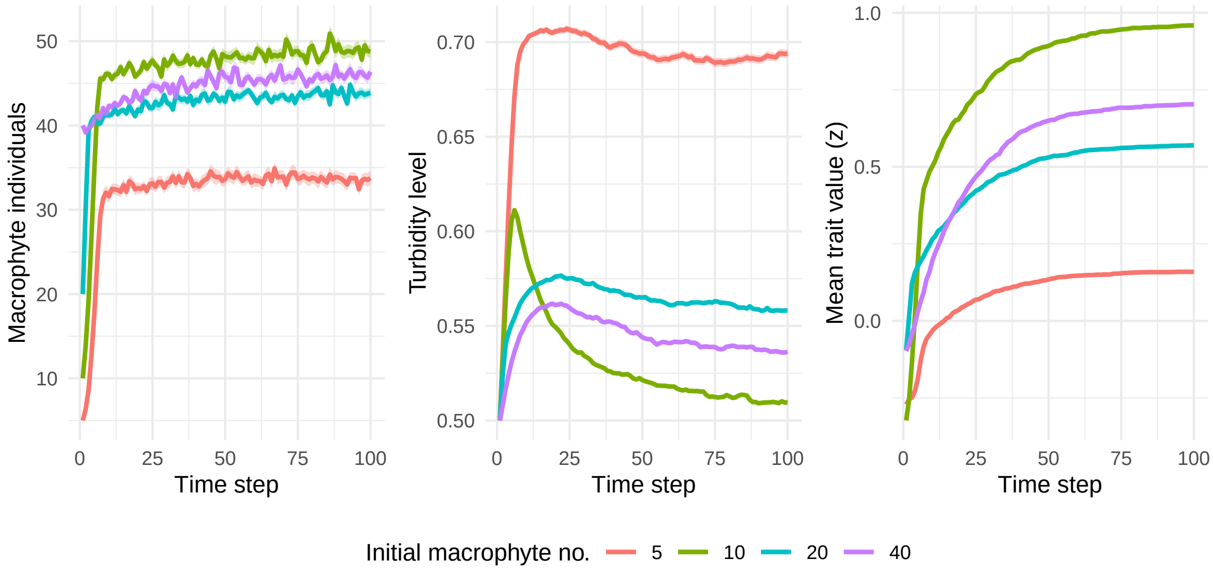

Figure 3: Plot of Macrophyte population size, turbidity level and trait value over time for differing initial population size. Solid lines indicated an average value (n = 30) at each time.