Simulations of an impulsive model for the growth of fruit trees

Theme 08 - Introduction to Systems Biology

Reproduce research

Lisa J.B. Hu (414264)

Niek R. Scholten (388602)

Bio-Informatics BFV2

Institute of Life Science & Technology

Hanze University of Applied Sciences

Tsjerk Wassenaar

June 17, 2022

Simulations of an impulsive model for the growth of fruit trees

Theme 08 - Introduction to Systems Biology

Lisa J.B. Hu (414264)
Niek R. Scholten (388602)
Bio-Informatics BFV2
Institute of Life Science & Technology
Hanze University of Applied Sciences
Tsjerk Wassenaar
June 17, 2022

Abstract

Droughts have affected the agriculture in the mediterranean in an increasingly severe matter. The decrease in rainfall and the effects of global warming have taken their toll on many orchards, especially fruit producing ones. Since the water supply of these orchards is always artificial because of the aforementioned factors, dwindling water capacities in reservoirs is a serious issue. This study aims to provide an insight into the effects of different irrigation patterns on the growth of these fruit trees. Without a sustainable plan for irrigation, whole populations of fruit trees might perish under a critical water deficit.

Summary

Climate change is one of the most severe problems facing planet earth right now. As a result of this, increasingly crippling droughts have swept over the mediterranean. This study is based on research of the jujube tree and the growth of its fruit under different irrigation circumstances. The main issue is the amount of work required to test this growth under varying conditions, so this paper tries to simplify the workflow for researching these scenarios using differential equations. These models result in vital information for growing patterns, as to better understand the effects of irrigation. An expansion on these models was also made to incorporate day and night cycles, this proved fruitful as the results drastically changed. The models were run using the deSolve package for R, and visualised using ggplot2.

List of Abbreviations

ODE Ordinary Differential Equation

Contents

1 Introduction

1.1 Purpose

The effects of climate change are an existential threat to planet earth. This paper's purpose is to reproduce and expand on the research done by E Duque-Marin to get a further understanding of the dynamics and different scenarios of fruit tree growth, and by extension shed light on a part of this humongous problem that is already cropping up around the world.

1.2 Theory

The growth of the fruit tree is highly impacted by the different variables in- and outside the tree. To construct an impulsive model, an order of assumptions must be made:

1. The growth dynamics is governed by an interaction between the variables energy, water, vegetative growth.
2. The fruit tree responds instantly to the irrigation application.
3. The model concerned an adult fruit tree (older than 5 years) with a suitable soil surface for the growth of the fruit tree.
4. Ideal agronomic management conditions.
5. Optimal environmental conditions: The energy of the system is constant.

The parameters concerning the model are summarized in Table 1.

Table 1: Parameters used for the model

Parameter	**Meaning**
q	Accumulated energy constant
r	Fruit trees intrinsic growth rate
N	Fruit carrying capacity
I	Irrigation water amount
β	Evapotranspiration rate
γ	Photosynthetic contribution rate
ω	Mortality rate of fruit trees

With that, the state variables can be denoted as following:

1. $E = E(t)$ the solar radiation at time t;
2. $W = W(t)$ the water amount in the soil at time t;
3. $C = C(t)$ the fruit biomass concentration at time t.

Under assumption 5, the state variables are reduced to only W and C. The variation of the amount of water in the system is denoted by $W'(t)$ and the variation in biomass is denoted by $C'(t)$. Considering there is no rainfall (p = 0), the water variation output is due to evapotranspiration at rate β. Besides that, water is crucial for promoting the growth of the fruit tree at rate r.

C'(t) corresponds to the water input at a rate r; in addition, it is also positively affected due to the water-energy-biomass growth interaction at a rate γ. There is an exit at an ω rate for the loss of natural death of the crop. This behavior applies to a continuous timescale (long duration), that governs the growth dynamics of the fruit tree. However, this model also presents a short timescale (pulse) in discrete time that represents the events in which water enters the system through irrigation supply.

The dynamics between the variables energy, amount of water, and concentration of biomass of the fruit tree are shown in Figure 1, adapted from [1]; where: E = Energy; W = Water amount; C = Fruit biomass concentration; r = Fruit trees intrinsic growth rate; N = Fruit carrying capacity; β = Evapotranspiration rate; γ = Photosynthetic contribution rate; ω = Mortality rate of fruit trees; p = Rainfall rate; I = Irrigation amount.

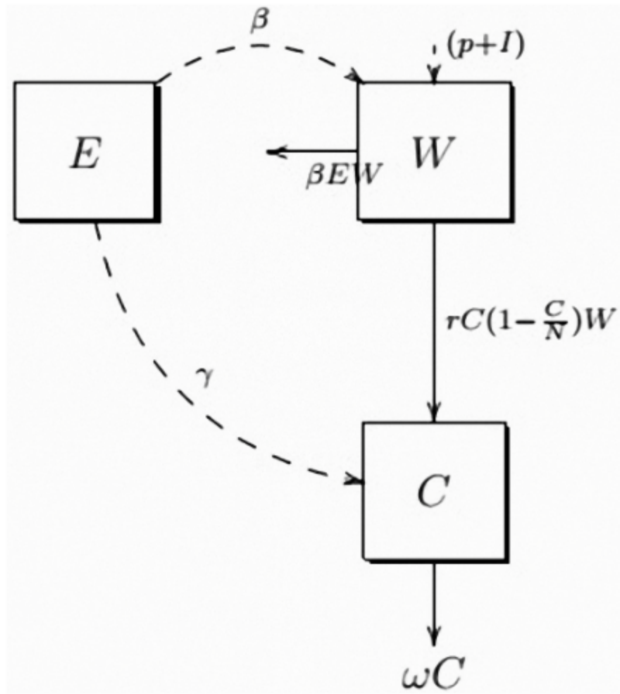

Figure 1: Diagram of dynamics

A model can be proposed that represents the growth dynamics of fruit trees, where the supply of irrigation is evidenced in the form of a pulse, and thus be described by impulsive differential equations 1.2. The differential equations count when $t \neq nT$ and the latter two count when $t = nT$; with the default being $nT = 8$.

Equations:

$$\begin{array}{ll} W'(t) = -\beta q W(t) - rC(t)(1 - \frac{C(t)}{N}W(t) & \\ C'(t) = rC(t)(1 - \frac{C(t)}{N})W(t) + \frac{\gamma q C(t) W(t)}{(C+1)(W+1)} - \omega C(1) & t \neq nT \\ \Delta W(nT) = I & \\ \Delta C(nT) = 0 & t = nT \end{array} \tag{1.2}$$

2 Materials and Methods

Since no datasets were used in the production of the results, the simulation data had to be generated. This data was plotted using R graphics.

2.1 Materials

The simulation data was obtained using the ODE (ordinary differential equation) function of the deSolve package. This function takes differential equations and parameters and calculates the output of these functions. Aforementioned equations can ben found in the theory section. Other packages like ggplot2, ggpubr, formatR and scales were used in data visualisation. (2)

Table 2: Software and packages

Software	**Package**	**Version**
R		4.0.4
	deSolve	1.32
	formatR	1.11.1
	ggplot2	3.3.5
	ggpubr	0.4.0
	scales	1.1.1

The simulation data consists of an index, the *time*, water level in the soil (W) and biomass (C). (3)

Table 3: The simulation data

	time	W	C
1	0.00000	0.600000	1.00000
2	1.00000	0.556209	1.02602
3	2.00000	0.514991	1.05076
4	3.00000	0.476281	1.07421
5	4.00000	0.440001	1.09637

2.2 Methods

The differential equations were translated into a mathematical model that deSolve could understand. The last two equations also needed to be incorporated; this was done through if/else statements checking the timestep. After several tests were run to ensure parity with the described model in the paper. See Appendix 7.2

After this, the model was updated to use one-hour steps instead of one-day steps. This lead to the model being adapted to incorporate different growth rates for day and night, for even more precise data. See Appendix 7.3.

Another few models were made to compare different irrigation intervals, as well as a model for watering the soil as soon as almost all water was drained from it. See Appendix 7.4 and 7.5.

3 Results

3.1 Replication

The growth of the fruit trees can be shown in multiple simulations, tweaking a few variables to see the differences between these. In Figure 2a, the courses of the two state variables are shown: The water amount in the soil (blue) and the growth of the fruit tree's biomass (red). Figure 2b shows what happens with the growth of the fruit tree if the parameter q changes, for $q = 0.1$ in red, $q = 0.5$ in black (the default value), and $q = 1$ in blue. The figures 2c and 2d conduct a delayed water irrigation in the model, with $nT = 16$.

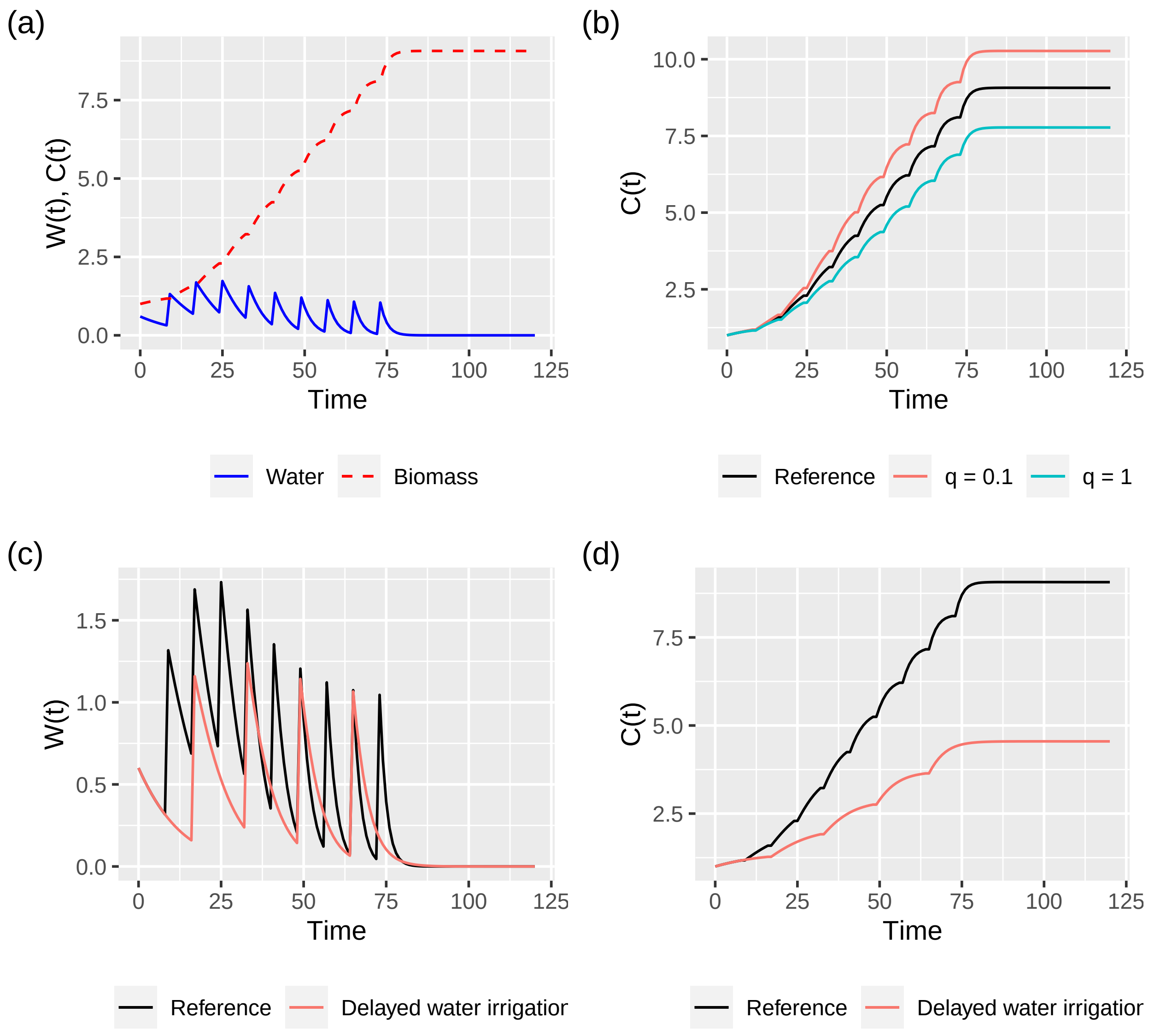

Figure 2: The different simulations

3.2 Expansion

The following simulations show the difference between the expanded model accounting for day and night cycles in red, and the original model from [2] in black. Figure 3 shows a decrease in overall biomass production, while steady growth is still retained. This change has to do with the fact that a little under half of a 24-hour period is calculated as “no growth” night-time.

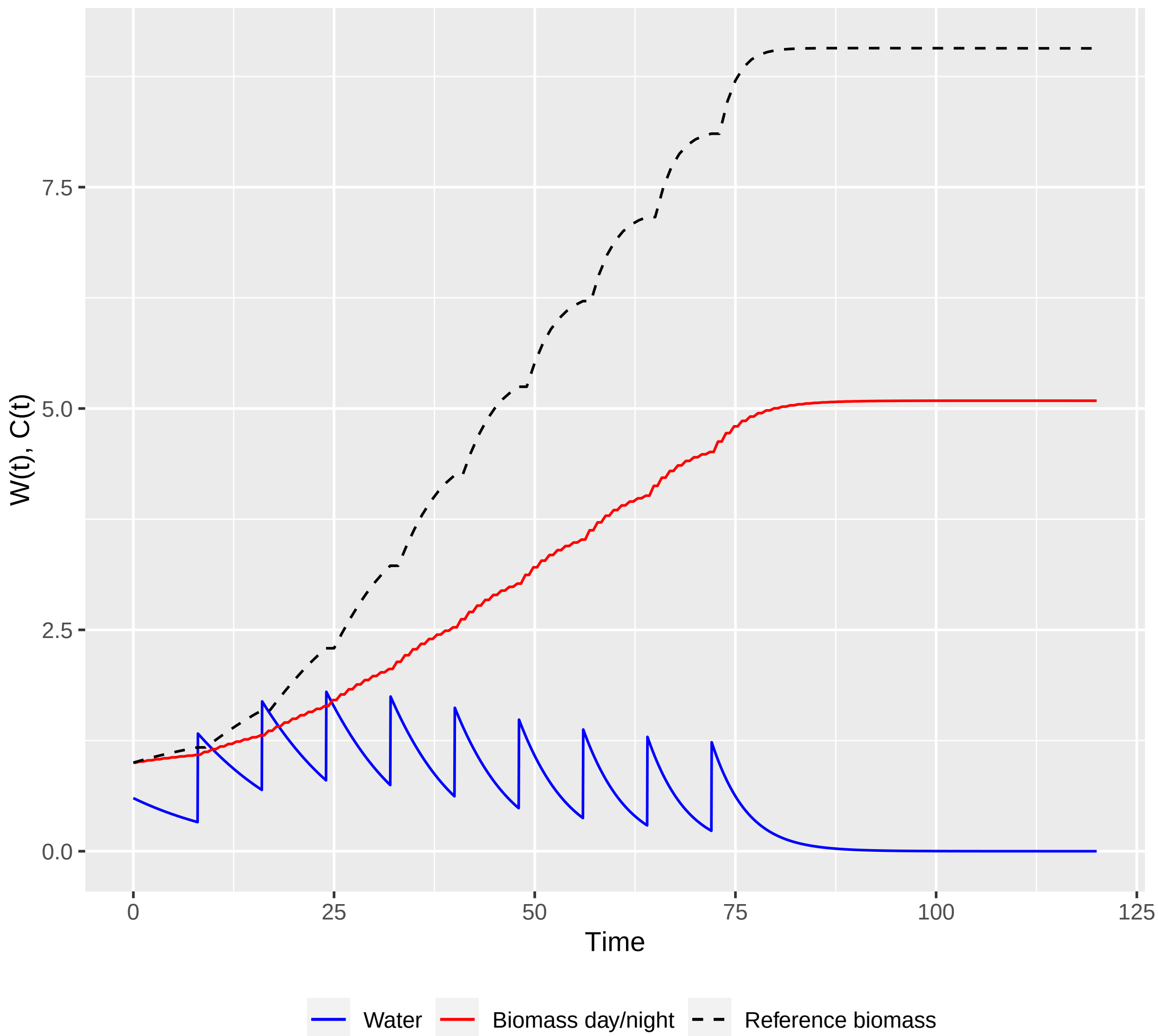

Figure 3: The expanded simulation

3.3 Comparison

Comparing different sets of irrigation data is vital to understanding the most efficient way of watering the fruit trees. Figure 4 shows the difference between the reference research from the paper and newly run simulations.

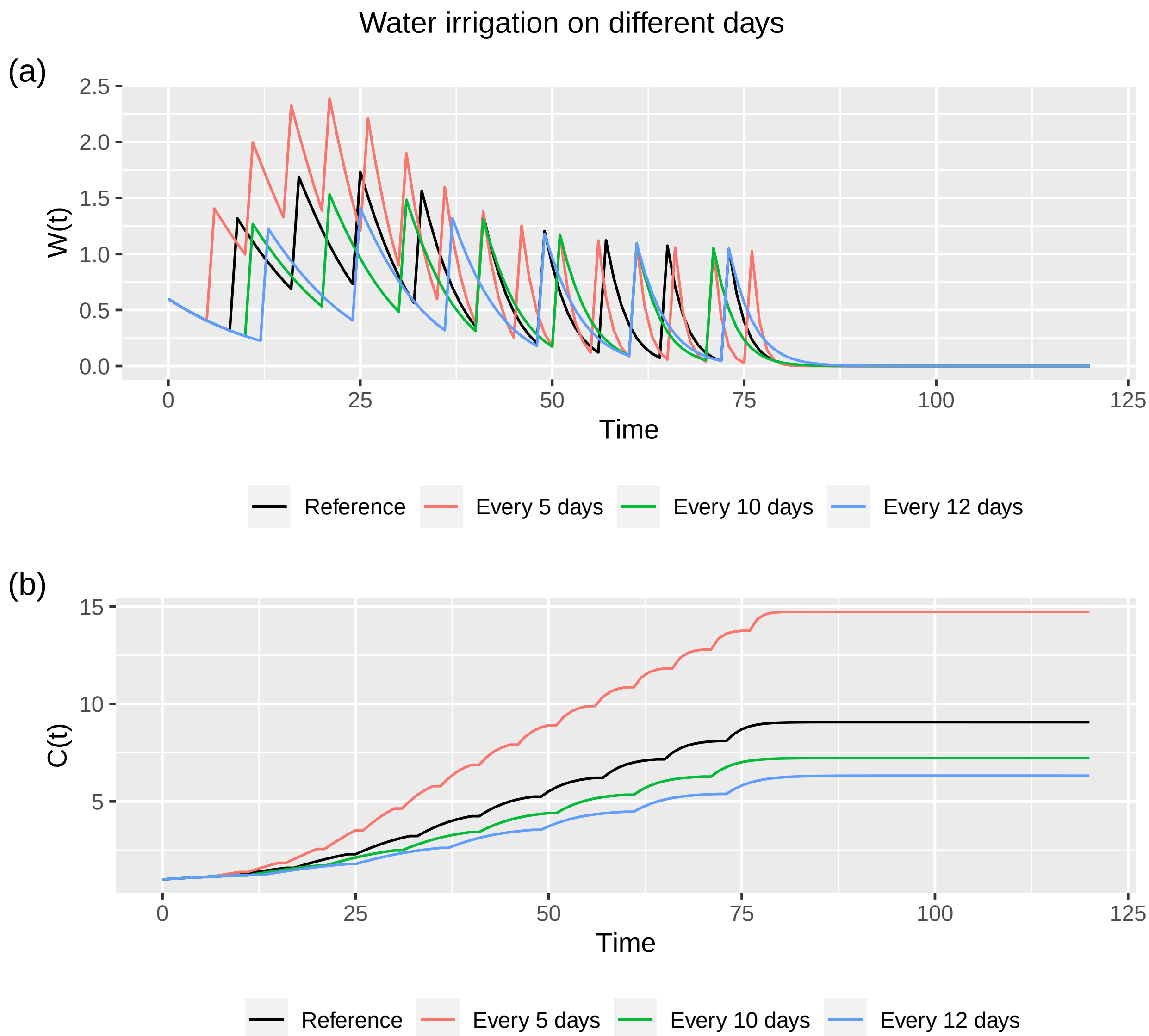

Figure 4: The simulation comparisons

3.4 Desiccation

Figure 5 shows a scenario in which the trees only gets water when the water level in the soil is critically low. This is a simulation of the worst case scenario, achieved through dynamically programming a steady state.

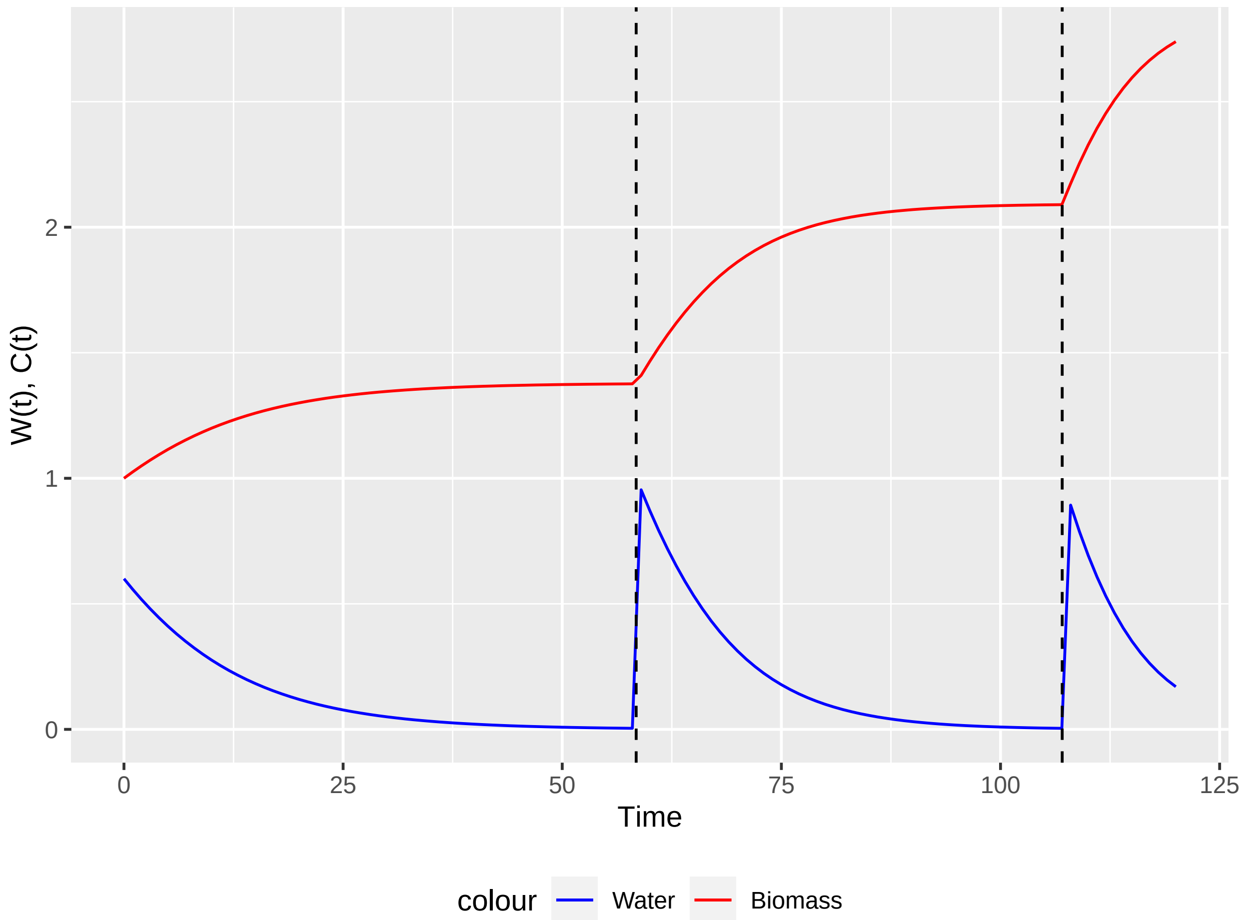

Figure 5: The edge case simulation

4 Conclusion

In this work, simulations of an impulsive system of nonlinear differential equations were reproduced to describe the growth of fruit trees when exposed the application of water by irrigation. These models were then expanded upon to incorporate day/night cycles for the trees to respond to. The results of this work not only exhibit the positive effects of a well scheduled irrigation system, but also the negative effects of a system that is not well scheduled. Results like these are very important factors in addressing problems that have to do with climate change induced drought. All this culminates into the conclusion that artificial irrigation is an essential part of growing fruit trees in areas with frequent drought and decreasing rainfall as an effect of climate change.

5 Discussion

- The day-night model does not conform assumption 5. Beside a non-constant energy output, the biomass should not grow in the evening, which should change the water loss for biomass to 0. Neither two were incorporated in the model.

6 References

1. E Duque-Marín *et al* 2021 *J. Phys.: Conf. Ser.* 2046 012017
2. E Duque-Marín *et al* 2022 *J. Phys.: Conf. Ser.* 2153 012018

7 Appendix

7.1 Appendix A

```r
## ---------------------------
##
## Name: functions.R
##
## Author: Lisa Hu
##
## Purpose: Script contains functions used in the result scripts
##
## Email: l.j.b.hu@st.hanze.nl
##
## ---------------------------

model <- function(t, y, parms){
  # Add water every given days, until day 80
  if(t %% as.numeric(parms["time"]) == 0 && t < 80 && t > 0){
    with(as.list(c(parms, y)), {
      dW <- I # I is the water irrigation
      dC <- 0 # There is no growth on those days
      return( list( c(dW, dC) ) )
    })
  }
  # Else the model runs with the equations
  else{
    with(as.list(c(parms, y)),{
      dW <- (-B * q * W) - (r * C * (1 - ( C/N ) ) * W)
      dC <- (r * C * (1 - ( C/N ) ) * W) + ( (g*q*C*W)/(C+1)*(W+1) ) - o * C
      return( list( c(dW, dC) ) )
    })
  }
}

day.night_model <- function(t, y, parms){
  # Add water every given days, until day 80
  if(t %% as.numeric(parms["time"]) == 0 && t < 80 && t > 0){
    with( as.list( c(parms, y)), {
      dW <- I * 24 # I is the water irrigation
      # NOTE : x24 because the water amount should not change
      dC <- 0 # There is no growth on those days
      return( list( c(dW, dC) ) )
    })
  }
  # Else the model runs with the equations
  else{
    if(t %% 1 <= 0.25 | t %% 1 >= 0.83){
      # During the night (no sun)
      with( as.list (c(parms, y)), {
        dW <- ((-B * q * W) - (r * C * (1 - ( C/N ) ) * W)) # Normal water drop
        dC <- 0 # No growth
        return( list( c(dW, dC) ) )
        })
```

```r
    }
    else{
      # During the day (sun)
      with( as.list( c(parms, y)),{
      dW <- ((-B * q * W) - (r * C * (1 - ( C/N ) ) * W))
      dC <- ((r * C * (1 - ( C/N ) ) * W) + ( (g*q*C*W)/(C+1)*(W+1) ) - o * C)
      return( list( c(dW, dC) ) )
      })
    }
  }
}

water_model <- function(t, y, parms){
  # Add water every given days, until day 80
  with(as.list(c(parms, y)),{
    dW <- (-B * q * W) - (r * C * (1 - ( C/N ) ) * W)
    dC <- (r * C * (1 - ( C/N ) ) * W) + ( (g*q*C*W)/(C+1)*(W+1) ) - o * C
    dI <- 0
    return( list( c(dW, dC, dI) ) )
  })
}

## Function to create plots
create.plots <- function(plot.values, ref.data, change.data){
  #' plot.values = The column name of the datas
  #' ref.data = The reference data
  #' change.data = The data that contains changed values
  data.names <- names(change.data)
  # Create colours for the different lines (except the reference data)
  colours <- hue_pal()(length(change.data))
  # y.val inserts the plot.value for the corresponding row of data.values
  y.val <- data.values[plot.values,]
  # The plot
  plt <- ggplot(data = ref.data, mapping = aes(x = time, y = !!sym(y.val$name) ) ) +
    # Lines (Reference data stays black)
    geom_line(aes(color = "Reference")) +
    unlist( mapply(function(single.data, data.name)
                        geom_line(data = single.data, aes(color = data.name) ),
                   change.data, data.names ) ) +
    # Labels
    labs(x = "Time", y = y.val$ylabel) +
    theme(legend.position = "bottom") +
    # Line colours
    scale_colour_manual(values = c("black", colours),
                        limits = c("Reference", names(change.data) ) ) +
    # Legend correction
    guides(color = guide_legend(title = ""))
  return(plt)
}

## Labels and titles for according value
data.values <- data.frame(name = c("W", "C"),
                          ylabel = c("W(t)", "C(t)"))
```

```
rownames(data.values) <- data.values$name
```

7.2 Appendix B

```r
## ---------------------------
##
## Name: results1.R
##
## Author: Lisa Hu
##
## Purpose: Script creates the first results for the final report
##
## Email: l.j.b.hu@st.hanze.nl
##
## ---------------------------

## ODE values
parameters <- c(q = 0.5, r = 0.043, N = 3000, I = 1,
                B = 0.06, g = 0.001, o = 0.00001, time = 8)
state <- c(W = 0.6, C = 1)
times <- seq(0, 120, by = 1)

## Run the simulations
ref.data <- as.data.frame(ode(times = times, y = state,
                              parms = parameters, func = model, method = "euler"))

## Determine the different q values
q.values <- list("q = 0.1" = 0.1,
                 "q = 1" = 1)

for(i in seq_along(q.values)){
  parameters$q <- q.values[[i]]  # Set new q value
  # Run the simulation and store in q.values
  q.values[[i]] <- as.data.frame(ode(times = times, y = state,
                                     parms = parameters, func = model, method = "euler"))
}

## Simulation for delayed water irrigation (every 16 days)
parameters <- c(q = 0.5, r = 0.043, N = 3000, I = 1,
                B = 0.06, g = 0.001, o = 0.00001, time = 16)

delay.data <- as.data.frame(ode(times = times, y = state, parms = parameters,
                                func = model, method = "euler"))
delay.data <- list("Delayed water irrigation" = delay.data)


## Create the plots
# The model simulation
plt1 <- ggplot(ref.data, mapping = aes(x = time)) +
          # The different lines
          geom_line(mapping = aes(y = W, color = "Water")) +
          geom_line(mapping = aes(y = C, color = "Biomass"), linetype = "dashed") +
          # Labels
          labs(x = "Time", y = "W(t), C(t)") +
          # Line colours
          scale_colour_manual(values = c("blue", "red"),
```

```
                              limits = c("Water", "Biomass")) +
          # Make the line of the Biomass a dashed line in the legend
          guides(color = guide_legend(title = "",
                                      override.aes = list(linetype = c(1, 2))))

# Different q values
plt2 <- lapply("C", create.plots, ref.data, q.values)

# Delayed water model
plt3 <- lapply(c("W", "C"), create.plots, ref.data, delay.data)

## Add figure annotation
plot.list <- append(list(plt1), c(plt2, plt3))
plot.tags <- c("(a)", "(b)", "(c)", "(d)")

for(i in seq_along(plot.list)){
  plot.list[[i]] <- plot.list[[i]] + labs(tag = plot.tags[i])
}

## Arrange plots
my.grid <- ggarrange(plotlist = plot.list, ncol = 2, nrow = 2,
                     common.legend = FALSE, legend = "bottom")
## Print plots
print( annotate_figure(my.grid) )
```

7.3 Appendix C

```r
## ---------------------------
##
## Name: results2.R
##
## Author: Lisa Hu
##
## Purpose: Script creates the day/night results for the final report
##
## Email: l.j.b.hu@st.hanze.nl
##
## ---------------------------

## ODE values
parameters <- c(q = 0.5, r = 0.043, N = 3000, I = 1,
                B = 0.06, g = 0.001, o = 0.00001, time = 8)
state <- c(W = 0.6, C = 1)
times <- seq(0, 120, by = 1/24)

## Run the simulations
d.n_data <- as.data.frame(ode(times = times, y = state, parms = parameters,
                              func = day.night_model, method = "euler"))

## Create the plot
ggplot(d.n_data, mapping = aes(x = time)) +
       # The different lines
       geom_line(mapping = aes(y = W, color = "Water")) +
       geom_line(mapping = aes(y = C, color = "Biomass day/night")) + # day/night
       geom_line(mapping = aes(y = C, color = "Reference biomass"),
                 data = ref.data, linetype = "dashed") + # Default model
       # Labels
       labs(x = "Time", y = "W(t), C(t)") +
       # Line colours
       scale_colour_manual(values = c("blue", "red", "black"),
                           limits = c("Water", "Biomass day/night",
                                      "Reference biomass")) +
       guides(color = guide_legend(title = "",
                                   override.aes = list(linetype = c(1, 1, 2)))) +
       # Theme
       theme(legend.position = "bottom")
```

7.4 Appendix D

```r
## ---------------------------
##
## Name: results3.R
##
## Author: Lisa Hu
##
## Purpose: Script creates the results for different days of water irrigation
##
## Email: l.j.b.hu@st.hanze.nl
##
## ---------------------------

## ODE values
parameters <- c(q = 0.5, r = 0.043, N = 3000, I = 1,
                B = 0.06, g = 0.001, o = 0.00001, time = 8) # time = 8 for reference
state <- c(W = 0.6, C = 1)
times <- seq(0, 120, by = 1)

## Run the simulations
ref.data <- as.data.frame(ode(times = times, y = state,
                              parms = parameters, func = model, method = "euler"))

time.values <- list("Every 5 days" = 5,
                    "Every 10 days" = 10,
                    "Every 12 days" = 12)

for(i in seq_along(time.values)){
  parameters$time <- time.values[[i]]  # Set the time value
  # Run the simulation and store in time.values
  time.values[[i]] <- as.data.frame(ode(times = times, y = state,
                              parms = parameters, func = model, method = "euler"))
}

## Create the plots
plts <- lapply(c("W", "C"), create.plots, ref.data, time.values)

## Add the figure annotations
plot.tags <- c("(a)", "(b)")
for(i in seq_along(plts)){
  plts[[i]] <- plts[[i]] + labs(tag = plot.tags[i])
}

## Arrange the plots
my.grid <- ggarrange(plotlist = plts, ncol = 1, nrow = 2,
                     common.legend = FALSE, legend = "bottom")
## Print the plots with a title
print( annotate_figure(my.grid,
                       top = text_grob("Water irrigation on different days") ) )
```

7.5 Appendix E

```r
## ---------------------------
##
## Name: results4.R
##
## Author: Lisa Hu
##
## Purpose: Script adds water to the system when it's 0
##
## Email: l.j.b.hu@st.hanze.nl
##
## ---------------------------

## ODE values
parameters <- c(q = 0.5, r = 0.043, N = 3000, B = 0.06, g = 0.001, o = 0.00001, time = 8)
state <- c(W = 0.6, C = 1, I = 0)
times <- seq(0, 120, by = 1)

## Determine what the root is
root <- function(t, y, parms){
  return(y["W"] - 4e-3)
}

## When root found, execute event
eventfun <- function (t, y, parms){
  y["I"] <- 1
  y["W"] <- y["W"] + y["I"]
  return(y)
}

## Run the simulation with events
sim.data <- ode(times = times, y = state, parms = parameters,
                func = water_model, rootfunc = root,
                events = list(func = eventfun, root = TRUE, terminalroot = 2))
roottimes <- attributes(sim.data)$troot  # Timesteps where root was found
sim.data <- as.data.frame(sim.data)

## Create plot
ggplot(sim.data, mapping = aes(x = time)) +
       # The different lines
       geom_line(mapping = aes(y = W, color = "Water")) +
       geom_line(mapping = aes(y = C, color = "Biomass")) +
       # Vertical lines where root was found
       unlist( mapply( function(x){
         geom_vline(xintercept = x, linetype = "dashed")
                                   }, roottimes) ) +
       # Labels
       labs(x = "Time", y = "W(t), C(t)") +
       # Line colours
       scale_colour_manual(values = c("blue", "red"),
                           limits = c("Water", "Biomass")) +
       # Theme
       theme(legend.position = "bottom")
```