Caroline Schulte

Professor Waller

BE 523

April 3, 2020

Project Report 2

Problem Description

For this project, my group and I are focusing on creating organic photovoltaic greenhouse solar models as well as biological models of tomato and lettuce crop growth. Ideally, the information derived from the OPV models - solar model and shading model - will be used in the crop growth models to predict cultivation outputs for tomato and lettuce crops. For my specific section, I will be modeling tomato growth. The tomatoes being used are Rebelski - a variety of beef tomato. The two most important factors we are considering with regards to plant growth are temperature and light inputs. Both factors influence photosynthesis, which in turn affects plant yields.

The general equation for photosynthesis is $6H_2O\ +\ light\ +\ 6CO_2 \ –> C_6H_{12}O_2\ +\ 6O_2$. Essentially, carbon dioxide and water in the presence of light is absorbed by plants to produce glucose and oxygen. Oxygen is created as a bi-product, but glucose is used by plants for growth and fruit bearing. Light received by plants is generally given in daily light integral (DLI). DLI is the total amount of photosynthetically active radiation (400-700nm) received per square meter per day. This range of light is important because it is the only light energy that plants respond to. For example, blue light (400-500nm) affects leaf growth, red light (600-700nm) affects flowering, far-red light (700-700nm) speeds up conversion of a red and far-red light photoreceptor allowing the plants to produce greater yields. Additionally, the chlorophyll within the plants can absorb PAR energy in order to create food. In general, plant growth rate per day is approximately linear to the DLI meaning plant growth increases with increasing average DLI received by plants. Therefore, knowing the DLI plants are receiving can be extremely helpful in predicting plant growth. Similarly, photosynthetic photon flux density (PPFD) is also helpful for predicting plant growth as it is a measure of the amount of photons falling on a particular surface each second. Ideally, tomatoes will receive a PPFD of 185 $\mu mol/s*m^2$.

With temperature, as it increases, photosynthesis, transpiration, and respiration rates increase. Consequently, growth rate tends to increase with a rise in temperature and decrease with a drop in temperature. However, once the temperature exceeds around 30 ^0C, the rate of photosynthesis slows down because the enzymes involved in the chemical reactions are destroyed. Additionally, when temperatures are too high, the air is able to accommodate more water vapor. This leads to plants transpiring excessively, causing them to be stressed as they are unable to replace the amount of water lost. Tomatoes specifically thrive in temperatures of about 21^0C during the day and night temperatures between 60 and 64 ^0C.

Original Model Overview

The model I will be making will be in Python; it will be derived from a simplified TOMGRO model as well as the original TOMGRO model. The TOMGRO model was made in 1991 as a means of modeling tomato growth and yield within a controlled environment. The inputs for the model were temperature, CO2, and photosynthetic photon flux (PPFD). The output of the model included 69 state variables (Jones, James...). On the contrary, the reduced TOMGRO model only has five variable outputs. My model will specifically focus on plant development through node number quantification and leaf area index (LAI) from the reduced model. I will also be modeling the number of fruits grown using the original model.

Node development rate is expressed using the equation $\frac{dN}{dt} = N_m * f_N(T)$ (see table one for definition of variables). This gives the number of nodes on the main stem developed per day. For f_N(T), the equation assumes a single optimal temperature applying an increasing or decreasing factor (V0 = 0.25+0.25T and V1 = 2.5-0.05T) to simulate plant response to variations in air temperature (Bacci). In this equation, N_m is estimated; for this project it is estimated to be 0.02083 nodes/day based on tomato phenology (Shamshiri, Redmond).

LAI is given as a function of time. The model equation is meant to express the leaf area developed (m^2[leaf]/m^2[ground]) per day. If LAI is less than or equal to LAI maximum, the model equation is $\frac{d(LAI)}{dt} = \rho * \delta * \lambda(T_d) * \frac{exp[\beta*(N-N_b)]}{1+exp[\beta*(N-N_b)]} * \frac{dN}{dt}$. Otherwise, $\frac{d(LAI)}{dt}$ = 0. In the LAI equation, β, δ , and N_b are all estimated values. Table two shows the possible range of values that these can be. For purposes of this model, λ (T_d) is unnecessary because temperatures in the greenhouse will remain fairly constant meaning this will not have consequential effects on the leaf area expansion simulations (Shamshiri, Redmond).

To model number of fruit, the net rate of change of the number of fruit in a particular age class is given by the following equation

$\frac{dN_F(i)}{dt} = r_F(T) * F(C) * n_F * N_F(i-1) - r_F(T) * F(C) * n_F * N_F(i) - P_F(i)$. This gives the number of fruit per meter squared per day. The variables can be found defined in table one below.

In order to create this model, I am breaking the process down into five parts. The first is establishing a model of the growth rate response to temperature using the node development rate equation. The equation assumes a maximum value of 30 ^{0}C and zero values at -10 ^{0}C and 50 ^{0}C ; these values will be the graph's boundaries. Secondly, I will use this graph to create a user interface that allows a user to enter a temperature in order to receive an output of growth rate in terms of nodes/day. For LAI, using the temperature that was previously entered as well as the equation for LAI, a model of LAI as a function of days since transplanting will be shown. The user can then enter days since transplanting and receive output of an approximate LAI. The last part has to do with using the amount of fruit. The program will use the temperature value obtained and export a graph depicting the number of fruit as a function of time. Once again, a user can input the days since they planted crops in order to find out the average amount of fruit.

Variable	**Definition**	**Units**
ρ	Plant density	No. [plants] * m^{-2}[ground]
δ	Maximum leaf area expansion per node	m^2[leaf] * $node^{-1}$
β	$node^{-1}$	Coefficient in expolinear equation (node)
λ (T_d)	Temperature function to reduce rate of leaf area expansion	Unitless (0 to 1)
$f_N(T)$	Function to modify node development rate as a function of hourly temperature min(1,min(0.25+0.25T, 2.5-0.05T)) with T in degrees celsius	Unitless (0 to 1 function)

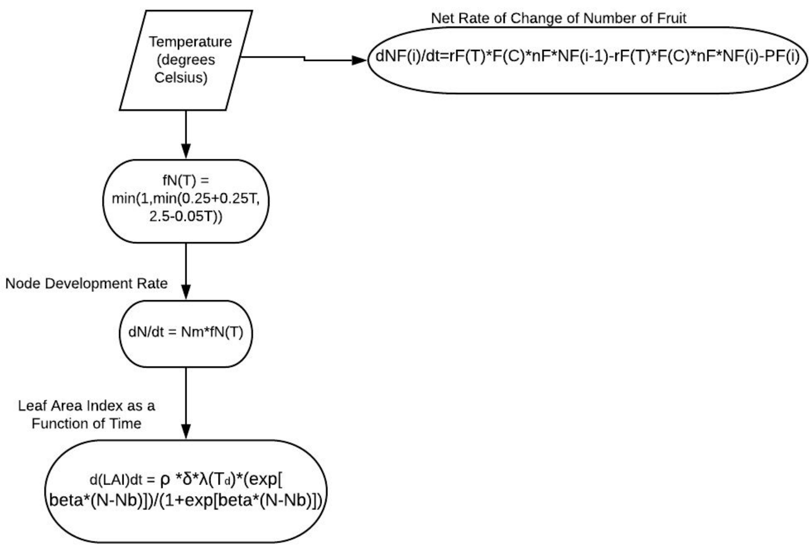
N	Number of nodes on mainstem	No. of nodes
N_b	Coefficient in expolinear equation, projection of linear segment of LAI vs N to horizontal axis	node
N_m	Maximum rate of node appearance (at optimal temp)	node * d^{-1}
T	Hourly temperature	^0C
$r_F(T)$	Rate of development of fruit at temperature T and 350 ppm CO2 concentration	1/d
F(C)	Scaler function of CO2 concentration that modifies the rate of development for CO2 levels above or below 350 ppm	unitless
n_F	Number of fruit age classes	No. fruit
N_F	Number of fruit/m^2 for age class i	fruit/m^2
P_F	Fruit loss caused by shading, insect damage, or diseases	No. fruit

Table 1: Defined variables used in model equations.

Parameter	Description	Value	Range of Estimation	Values Reported by Other Authors
N_b	Parameter in expolinear equation	13* (16)	8-25	16
δ	Maximum leaf area expansion	0.041* (0.038)	0.01-0.1	0.030
β	Parameter in expolinear equation	0.22* (0.169)	0.06-0.5	0.169

Table 2: Parameters estimated for reduced TOMGRO growth model (Ramirez, Shamshiri, Redmond)*

Original Model Flowchart



Adjustments to the Original Proposal

In the original proposal, the idea was that reduced TOMGRO equations would be used to code the tomato growth model. However, this was found to be extremely complex. Thus, the conclusion was to only use the TOMGRO code for node development. Then, based on assumptions, new equations were made to determine growth of the head of the plants as well as the number of trusses and flowers grown.

Description of Actual Program

The program I coded has 6 different functions. The functions utilize 6 inputs from the user (ideally, these values would be outputs from Bekah's code). These inputs are average daily temperature for the OPV and non-OPV sections in °C, the average DLI values for the OPV and non-OPV sections in mol/m^2/day, and the number of days the plant has been growing (this assumes the days are after vegetative growth has occurred, meaning the plant is producing flowers). This program also only outputs the values for one plant from each section over time (OPV vs Non-OPV).

The first function (DLI) is used to calculate the effect of DLI on the growth of the tomatoes. To make this function, I followed the same criteria used in TOMGRO to modify development based on temperature inputs. Essentially, I assumed that the ideal DLI values for tomatoes were between 22 and 30 mol/m^2/day (Torres). Then, I assumed that the highest DLI the plants could intake was 40 mol/m^2/day, and the lowest was 10 mol/m^2/day ("Typical..."). For these, I added 5 to 40 and subtracted 5 from 10 for the parameters so 10 and 40 mol/m^2/day had an effect greater than zero. To explain, to calculate the effect of the DLI values, I took the equation 1 - x (DLI-30) = 0 and solved for x given certain parameters. I used this equation assuming 30 mol/m^2/day was the ideal DLI. For example, for the lower effect (DLI between 22 and 5), I solved for x using 5 as the DLI value. This resulted in x = - 0.04. Thus, the equation for DLI between 22 and 5 mol/m^2/day was 1+0.04*(DLI-30). I performed this analysis for each section, with the sections being 22 - 30 mol/m^2/day, 30 - 45 mol/m^2/day, and 5 - 22 mol/m^2/day. The output from this function is an integer between 0 and 1 representing the effect average DLI will have on growth.

The second function (F_N) is for modifying node development given the average temperature. This was taken directly from the TOMGRO model (Gyosit), but I altered the values. The ideal temperature range for Rebelski tomatoes is between 20 and 22 degrees Celsius (“Rebelski…”). The maximum temperature was assumed to be 35 °C and the lowest to be 10 °C (Editors...). In this function, I used the same process to solve for the equations for each section of temperatures that I used for the DLI function, with the sections being 20 - 22 °C, 22 - 35 °C, and 10 - 20 °C. The output from this function is an integer between 0 and 1 representing the effect that temperature will have on growth.

The next function (dNdt) is used to determine the change in node development over time, with time being in days. In this function there is a constant. For the OPV section, the constant is 3/7. The constant comes from the assumption that the plants grow 1 foot per week (Rorabaugh), meaning three nodes are developed per week hence the 3/7 (three nodes per week or 0.429 nodes per day). For the non-OPV section, the constant Nm is 4.5/7 (assuming there is 1.5 feet of growth in a week equating to 4.5 nodes per week or 0.643 nodes per day). Then, the equation dN/dt = Nm * FN * DLI is used to calculate an output. This equation takes into account the outputs from functions one and two. To explain, if the DLI is ideal (30 mol/m^2/day), and the temperature is ideal (22 °C), then the constants FN and DLI will be 1, meaning the growth dN/dt will be ideal at 3/7 or 4.5/7 (depending on if you are looking at the OPV or non-OPV section). The output of this function is an integer giving the nodes developed on a plant over time.

The next function (Nodes_developed) is used to calculate how many nodes were developed each day for the amount of days input by the user. This is used incorporating the same equation from the previous function, but the output is an array so a graph can be shown plotting the sum of nodes developed over the amount of days the plant has been growing.

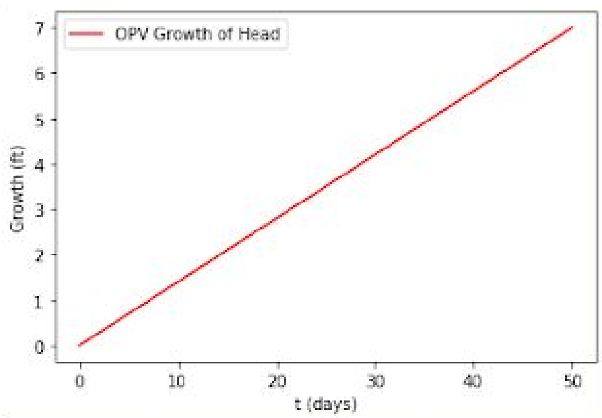
The next function (Flowers_developed) is used to show how many flowers have developed on a plant. This is based on the assumption that there is one truss developed per three nodes developed, and that three flowers develop on each truss (Rorabaugh). Consequently, the function takes in the number of nodes that have developed, and divides this by three to get the number of trusses. Then, the number of trusses is multiplied by three to find how many flowers have grown. The output is an array given how many flowers and trusses have been developed over the days of growth.

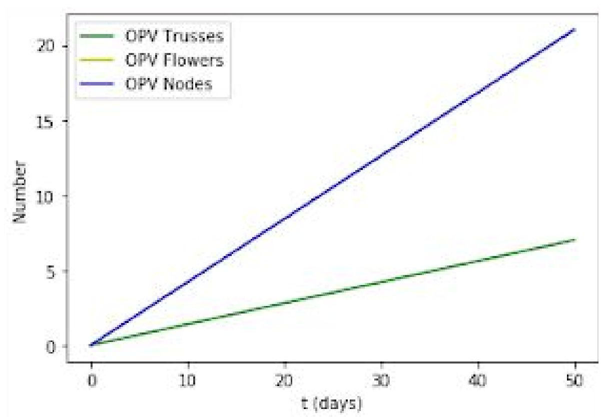
The last function (Head_Growth) deals with the growth of the head of the plant. This function assumed that one foot of growth occurs for every three nodes developed (Rorabaugh). Thus, the function takes the input of the array of nodes developed per day, divides the array by three, and returns how much the plant has grown per day.

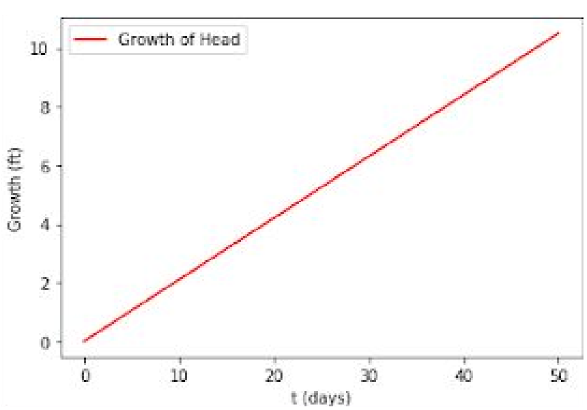
Graphs and Output

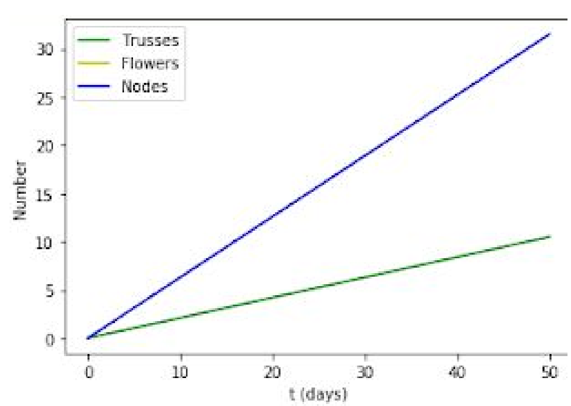
The program itself outputs four graphs. Two of the graphs represent the overall growth of the head of a plant over time - one is for the OPV section and one is for the non-OPV section. The other two graphs show how many flowers, trusses, and nodes have developed over time - one is for the OPV section and one is for the non-OPV section. Each graph has an x-axis of days, and the outputs plotted are sums. Below is an example of what one might expect in the output. With ideal temperatures, and DLI over 7 weeks, the OPV plant grows 7 ft and the non-OPV plant grows 10.5 ft.

```
temperature for OPV section (C) = 22

temperature for the non-OPV section is (C) = 22

Number of Days of Growth = 49

DLI for OPV section (mol/m^2/day) is = 30

DLI for non-OPV section (mol/m^2/day) is = 30
```

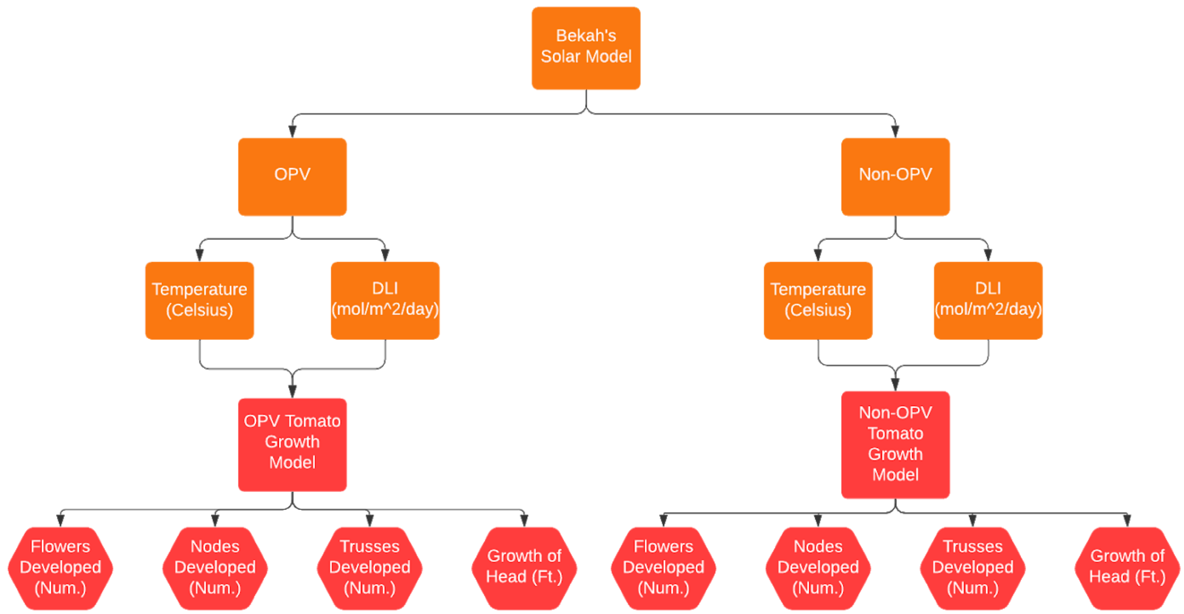








Actual Individual Flowchart



Code

```python
import numpy as np
import matplotlib.pyplot as plt

Temp_OPV = int(input("Temperature for OPV section (C) = "))
Temp = int(input("Temperature for the non-OPV section is (C) = "))
Days = int(input("Number of Days of Growth = "))

DLI_OPV_Input= int(input("DLI for OPV section (mol/m^2/day) is = "))
DLI_Input = int(input("DLI for non-OPV section (mol/m^2/day) is = "))


#print("temperature input for OPV section in degrees C is ", Temp_OPV)
#print("temperature input in degrees C for the non OPV section is ", Temp)

dt = 1 # time step (days)
N_t = int(Days/dt)
t = np.linspace(0,(N_t+1)*dt, N_t+1)



# OPV - covered
def DLI_OPV (DLI): # Function to calculate the effect of DLI on the growth of the plants given optimal DLI is 30 mol/m^2/day and minimum is 20 mol/m^2/day
    if(DLI >= 22 and DLI <= 30):
        DLI_effect = 1.0 + 0.05625 * (DLI - 30) # (0.55-1) / (22-30) = x = 0.05625
    elif(DLI > 30 and DLI < 45):
         DLI_effect = 1.0 - 0.066 * (DLI - 30) # 1 / (45 - 30) = x = 0.066
    elif(DLI < 22 and DLI > 5):
        DLI_effect = 1.0 + 0.04 * (DLI - 30) # - 1 / (5-30) = x = 0.04
    else:
        DLI_effect = 0
    #print(F_T)
    return(DLI_effect);

DLI_OPV_Effect = DLI_OPV(DLI_OPV_Input)
```

```python
def F_N_OPV(T): # Function to modify node development rate based on temperature
# This equation assumes that the ideal temp is 28 C for tomato growth

    if(T > 20 and T <= 22):
        F_T = 1.0 + 0.225 * (T - 22)  # (0.55 - 1) / (20-22) = x = 0.225
    elif(T > 22 and T < 35):
        F_T = 1.0 - 0.0455 * (T - 22) # 1 / (35-22) = x = 0.0769
    elif (T < 20 and T > 10 ):
        F_T = 1.0 + 0.0833 * (T-22) # -1 / (10 - 22) = x = 0.0833
    else:
        F_T = 0
    #print(F_T)
    return(F_T);

F_T_OPV = F_N_OPV(Temp_OPV) # The output from the function to modify node development rate

#print("Function to modify node development for OPV is", F_T_OPV)


def dNdt_OPV(FN, DLI): # The function for change in node development versus time using the output from the first function
  Nm = (3/7) # Constant for maximum rate of node appearance (node/day) given 3 nodes develop per week
  dN_dt = Nm * FN * DLI
  return (dN_dt);

dN_dt_OPV = dNdt_OPV(F_T_OPV, DLI_OPV_Effect) # Calling node development function
#print ("Rate of node development for OPV = ", dN_dt_OPV)


def Nodes_developed_OPV(FN, DLI): # Nodes developed given F_T and how many weeks have passed
    N_m = (3/7) # constant. 3 nodes per 1 foot of growth
    N = np.zeros(N_t+1)
    for n in range(1, N_t + 1):
        N[n] = N[n-1]+ (N_m*FN*DLI)*dt
    return N

Nodes_OPV = Nodes_developed_OPV(F_T_OPV, DLI_OPV_Effect) #input the number of days
#print ("Number of nodes developed for OPV is = ", Nodes_OPV)
```

```python

def Flowers_developed_OPV(Num_nodes):
    T = Num_nodes/3 # Number of trusses
    F = T * 3 # Number of flowers
    return(F, T)

Flowers_OPV, Trusses_OPV = Flowers_developed_OPV(Nodes_OPV)
#print("flowers and trusses developed for OPV are", Flowers_OPV, Trusses_OPV)


def Head_Growth_OPV(Num_nodes):
    Growth = (Num_nodes/3) # 3 nodes for every 1 foot of growth
    return(Growth)

Head_Growth_OPV_Total = Head_Growth_OPV(Nodes_OPV)
#print("The plant has grown this many feet under the OPV section= ", Head_Growth_OPV_Total)

# Graphs for OPV Section
plt.figure(1)
plt.plot(t,Head_Growth_OPV_Total,'r')
plt.legend(['OPV Growth of Head'], loc = 'upper left')
plt.xlabel('t (days)'); plt.ylabel('Growth (ft)')

plt.figure(2)
plt.plot(t,Trusses_OPV,'g', t, Flowers_OPV, 'y', t, Nodes_OPV, 'b')
plt.legend(['OPV Trusses', 'OPV Flowers', 'OPV Nodes'], loc = 'upper left')
plt.xlabel('t (days)'); plt.ylabel('Number')
```

```python
dt = 1 # time step (days)
N_t = int(Days/dt)
t = np.linspace(0,(N_t+1)*dt, N_t+1)

def DLI (DLI): # Function to calculate the effect of DLI on the growth of the plants given optimal DLI is 30 mol/m^2/day and minimum is 20 mol/m^2/day
   if(DLI >= 22 and DLI <= 30):
        DLI_effect = 1.0 + 0.05625 * (DLI - 30) # (0.55-1) / (22-30) = x = 0.05625
   elif(DLI > 30 and DLI < 45):
         DLI_effect = 1.0 - 0.066 * (DLI - 30) # 1 / (45 - 30) = x = 0.066
   elif(DLI < 22 and DLI > 5):
        DLI_effect = 1.0 + 0.04 * (DLI - 30) # - 1 / (5-30) = x = 0.04
   else:
        DLI_effect = 0
    #print(F_T)
   return(DLI_effect);

DLI_Effect = DLI(DLI_Input)

def F_N(T): # Function to modify node development rate based on temperature
# This equation assumes that the ideal temp is 28 C for tomato growth

    if(T > 20 and T <= 22):
        F_T = 1.0 + 0.225 * (T - 22)  # (0.55 - 1) / (20-22) = x = 0.225
    elif(T > 22 and T < 35):
        F_T = 1.0 - 0.0455 * (T - 22) # 1 / (35-22) = x = 0.0769
    elif (T < 20 and T > 10 ):
        F_T = 1.0 + 0.0833 * (T-22) # -1 / (10 - 22) = x = 0.0833
    else:
        F_T = 0
    #print(F_T)
    return(F_T);

F_T = F_N(Temp) # The output from the function to modify node development rate

#print("Function to modify node development for non-OPV is", F_T)


def dNdt(FN, DLI): # The function for change in node development versus time using the output from the first function
  Nm = (4.5/7) # Constant for maximum rate of node appearance (node/day) given 4.5 nodes develop per week because 1.5 ft growth per week
  dN_dt = Nm * FN * DLI
  return (dN_dt);
```

```python
dN_dt = dNdt(F_T, DLI_Effect) # Calling node development function using the output from the first function
#print ("Rate of node development for non-OPV = ", dN_dt)


def Nodes_developed(FN,DLI): # Nodes developed given F_T and how many weeks have passed
    N_m = (4.5/7) # constant. 3 nodes per 1 foot of growth
    N = np.zeros(N_t+1)
    for n in range(1, N_t + 1):
        N[n] = N[n-1]+ (N_m*FN*DLI)*dt
    return N

Nodes = Nodes_developed(F_T, DLI_Effect) #input the number of days
#print ("Number of nodes developed for non-OPV is = ", Nodes)


def Flowers_developed(Num_nodes):
    T = Num_nodes/3 # Number of trusses
    F = T * 3 # Number of flowers
    return(F,T)

Flowers, Trusses = Flowers_developed(Nodes)
#print("flowers and trusses developed for non-OPV are", Flowers, Trusses)


def Head_Growth(Num_nodes):
    Growth = (Num_nodes/3) # 3 nodes for every 1 foot of growth
    return(Growth)

Head_Growth_Total = Head_Growth(Nodes)
#print("The plant has grown this many feet = ", Head_Growth_Total)

# Graphs for Non-OPV
plt.figure(3)
plt.plot(t,Head_Growth_Total,'r')
plt.legend(['Growth of Head'], loc = 'upper left')
plt.xlabel('t (days)'); plt.ylabel('Growth (ft)')

plt.figure(4)
plt.plot(t,Trusses,'g', t, Flowers, 'y', t, Nodes, 'b')
plt.legend(['Trusses', 'Flowers', 'Nodes'], loc = 'upper left')
plt.xlabel('t (days)'); plt.ylabel('Number')
```

Conclusion

Modeling tomato growth turned out to be extremely complex. Even the reduced TOMGRO model using 5 equations ended up incorporating an abundance of algorithms. In the future, I feel that it would be best to collect data from the greenhouse, and then develop the equations afterwards based on the data obtained. If I had more time, I would incorporate vegetative growth (growth before the flowers begin forming), and make the graphs cumulative for all of the plants in each section rather than only showing data for one plant in each section. I would also want the model to be able to take in average weekly DLI and temperature values, rather than just one total average value for each. This would make the growth models far more accurate, as temperature and DLI will be fluctuating over the various months and seasons.

References

Bacci L, Battista P, Rapi B. Evaluation and adaptation of TOMGRO model to italian tomato protected crops. *N Z J Crop Hortic Sci*. 2012;40(2):115-126. https://doi.org/10.1080/01140671.2011.623706. doi: 10.1080/01140671.2011.623706.

Editors of Publications International, Ltd. “Tomatoes.” *HowStuffWorks*, HowStuffWorks, 8 Mar. 2007, home.howstuffworks.com/tomatoes1.htm.

Gyosit. “TOMGRO.py.” *GitHub*, gist.github.com/gyosit/abeab4e595d7ddcd65b55c1270d240c8. “Rebelski Slicing Tomatoes.” *Johnny's Selected Seeds*, www.johnnyseeds.com/vegetables/tomatoes/slicing-tomatoes/rebelski-f1-tomato-seed-3017.html.

Jones, J. W., et al. “Reduced State-Variable Tomato Growth Model.” *Transactions of the ASAE*, vol. 42, no. 1, 1999, pp. 255–265., doi:10.13031/2013.13203.

Jones, James & Dayan, Engin & Allen, Leon & Keulen, & Challa, H.. (1991). A dynamic tomato growth and yield model (TOMGRO). Transaction ASAE 34 (1991) 2. - ISSN 0001-2351. 34. 10.13031/2013.31715.

Ramírez, A., et al. “Calibration And Validation Of Complex And Simplified Tomato Growth Models For Control Purposes In The Southeast Of Spain.” *Acta Horticulturae*, no. 654, 2004, pp. 147–154., doi:10.17660/actahortic.2004.654.15.

Rorabaugh, Patricia A. *Introduction to Hydroponics and Controlled Environment Agriculture*. Controlled Environment Agriculture Center, 2015.

Shamshiri, Desa Ahmad, et al. "Evaluation of the Reduced State-Variable TOMGRO Model Using Boundary Data." *2016 ASABE International Meeting*, 2016, doi:10.13031/aim.20162454205.

Shamshiri, Redmond R., et al. "Adaptive Management Framework for Evaluating and Adjusting Microclimate Parameters in Tropical Greenhouse Crop Production Systems." *Plant Engineering*, 2017, doi:10.5772/intechopen.69972.

Torres, Ariana & Lopez, Roberto & Floriculture, Purdue. (2012). Measuring Daily Light Integral in a Greenhouse.

"Typical PPFD and DLI Values per Crop." *MechaTronix Mastering LED Cooling for High Bay & Horticulture Lighting*, www.horti-growlight.com/typical-ppfd-dli-values-per-crop.