
SIMULATING THE GROWTH OF PLANTS: A SIMPLE GROWTH MODEL OF VARIOUS PLANTS AND THE EFFECT OF ENVIRONMENTAL FACTORS

A PREPRINT

 **Ryan Goss**

Department of Computer Science (CSCI 5314)
University of Colorado
Boulder, CO 80301
ryan.goss@colorado.edu

 **Pourna Sengupta**

Department of Computer Science (CSCI 4314)
University of Colorado
Boulder, CO 80301
pourna.sengupta@colorado.edu

May 16, 2021

ABSTRACT

A highly complex process in nature and the environment is the growth of plants and organic biomass. Understanding, simulating and predicting these processes as a confluence between biology and computer science has been a topic of intense research at least since the 1980s. This has spanned sub-fields of plant structural-functional modeling, plant-plant interactions to crop and macro-scale biomass prediction. Here, we present a simplified model for emulating the growth patterns of common greenhouse plants, accounting for environmental factors such as water and sunlight. Keeping sunlight distribution uniform throughout the greenhouse grid, we explore three different "greenhouse" watering scenarios containing up to three distinct plant species, each with their own water and sunlight sensitivity. With this framework, we show that a rudimentary simulation of plant growth (in terms of height and width) can be simulated and generally represents true-to-life plant growth behavior in response to these varying watering scenarios and plant presets (species).

Keywords plant growth · computational modeling · biomass simulation · dynamic modeling of plants

1 Introduction

A topic of intense study and interest in within the field of computational biological modeling is that of simulating the growth, structure and architecture of plants and biomass over time. While highly complex, such applications have been explored from a wide gamut of perspectives for the past decades. The topic areas receiving the most attention currently are perhaps structural architectural modeling of individual plant species (e.g., each anatomical stage of growth and corresponding physical structure), crop modeling to account for varying weather or environmental challenges and macro scale biomass modeling and prediction based on changing climate conditions, including in response to models of planetary climate change. Of course, while exploration is ongoing in many other related realms, we outline some of the previous related work in these topic areas, below.

1.1 Modeling Individual Plant Growth Structure

The field of individual plant structural growth modeling dates back several decades with aspirations of simulating each stage of plant growth and the associated behavior, structure and growth patterns. Given the breadth of the subject alone, the focus of such explorations have varied in terms of variables affecting growth (e.g., nutrients, oxygen/nitrogen levels, sunlight, plant species, cellular level modeling [1]) and approach (i.e., mathematical simulations [2], stochastic geometric growth modeling [3] and machine learning approaches mimicking real plant growth utilizing computer vision techniques [4]).

Most notably, several software approaches have been developed over time to best approximate individual plant growth. The first such incarnation dates back approximately two decades with software called Atelier de Modelisation de l'Architecture des Plantes (AMAP) which, in its early stages, was a rudimentary application for generating growth of various plants via some pre-programmed template controls while introducing some randomness to the model [5]. From this platform, a more modern plan architectural and growth modeling software program has been developed, called GreenLab, derived from AMAP [6]. It was developed as part of a cross-functional collaboration of botanists, physiologists and mathematicians. A multitude of different plant species and parameters are available to model growth dynamics, including nutrient content, competition of resources, sunlight, etc with the ability to monitor output variables and visualize growth with computer graphics. Some examples of the types of structural plant modeling created with GreenLab is shown in Figure 1.



Figure 1: Examples of simulated structural plant growth, created with GreenLab.

One such study utilizing GreenLab reported successful simulation of deterministic topological development of a Mongolian Scots pine species in its native habitat within a fragile and arid ecosystem. The authors observed accurate simulation of branching and growth patterns by using a stochastic functional-structural plant model in GreenLab 2 [7]. The authors reported the method to be a successful technique to accurately model for characterizing tree structure and biomass allocation, as shown in 2. A relatively recent publication argues that while the majority of plant modeling utilizes "process-based" modeling, which emulate factors contributing to growth, a "functional-structural plant model (FSPM)", like the Mongolian Scots pine model may be more flexible and definable with an ability to tweak factors ad hoc. Additionally, the authors, urge the need for a more generalizable FSPM that can be modularized and from a standard template for more efficient development [8].

1.2 Crop Biomass Modeling

A somewhat related popular research topic area to individual plant growth structural-functional modeling is that of crop or large-scale plant-matter modeling utilizing varying scales of geography. While the research realm deserves its own review paper, we provide a few recent primer examples. As accurate simulation of crop biomass holds promising applications ranging from yield prediction, planning for future food insecurities and deficits, optimization of farming

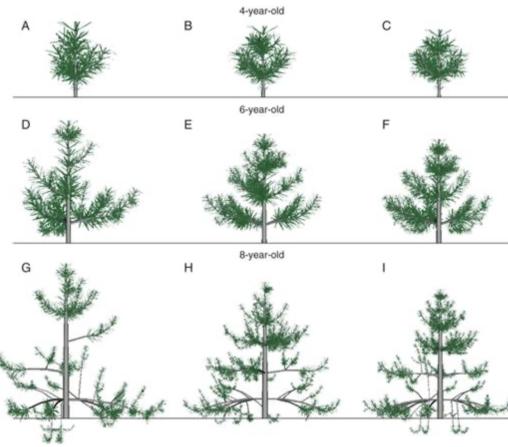
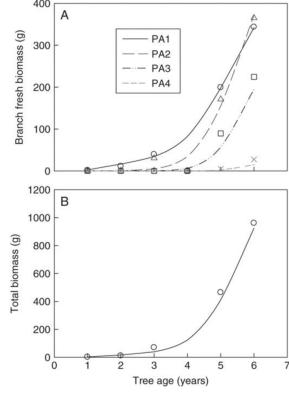
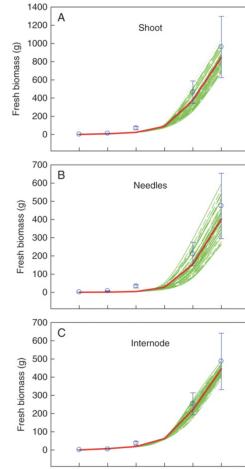
Fig. 6.**Fig. 4.****Fig. 5.**

Figure 2: Results from Wang et al. 2011, which modeled the growth of Mongolian Scots pine tree biomass and partitioning. "Fig 6" shows the continued modeled growth of trees at incremental ages. "Fig 4" graphs the branch and total biomass over aging of the tree. "Fig 5" shows the biomass of various tree organs from a monte carlo simulation (green lines are simulations, red line is average).

practices, planning for catastrophic events and informing policy decisions, the field has seen significant attention dating back several decades, with some of the first usable and somewhat predictive models dating back to the mid 1980s [9].

One such example of an early simulation explored the modeling of the maize mono-crop by integrating two previously reported crop models to increase the real-world predictive biomass. The authors demonstrated that previously developed models (termed CERES, a maize specific crop model and INTERCOM, a generic crop model used for maize) underestimated the biomass yield from real-world measured crop growth and developed a novel model borrowing from the strengths of the two previous models. Specifically, it utilized a temperature-driven phenological development, canopy integration of photosynthesis assimilates and organ-specific growth and respiration. The model requires less hybrid crop parameters as in the CERES model yet is able to account for hybrid variations of maize unlike the INTERCOM model. The authors demonstrated a dramatic improvement in crop biomass prediction using experimental results [10].

A more recent example of an approach to improve crop growth model accuracy was through the assimilation of satellite geospatial data. In an effort to improve the input parameters, so critical to the accurate prediction and modeling of crop growth models (CGM), researchers utilized a constellation of satellites (termed, "RedEye") to gather reflective spectra data and pigmentation data to be processed and interpreted to more accurately tweak input parameters. The updated input parameters are then delivered to the APSIM CGM to predict future biomass in the selected area. The authors determined a dramatic improvement in the predictive accuracy using this approach, indicating the utmost importance of correct input variables for modeling such a complex environmental process. Figure 3 shows the collection of spectra data with RedEye (left) and the APSIM biomass predictive mean (black) as compared to the measured value (green) from simulations of various locations (right) [11].

The importance of correct input variables was also emphasized in a recent publication demonstrating that collection of local survey data, such as planting time, climate conditions, fertilization, irrigation, etc. variables greatly enhances the predictive value for a CGM. Knowing and inputting these actual real-time variables were instrumental in significant increases in accuracy for forecasting biomass and food security of crop farms in Tanzania [9].

A very recent approach to improving CGM predictive accuracy, published in *Nature* this year, utilized modern machine learning techniques coupled with crop modeling. The study utilized both APSIM CGM input parameters (i.e., yield,

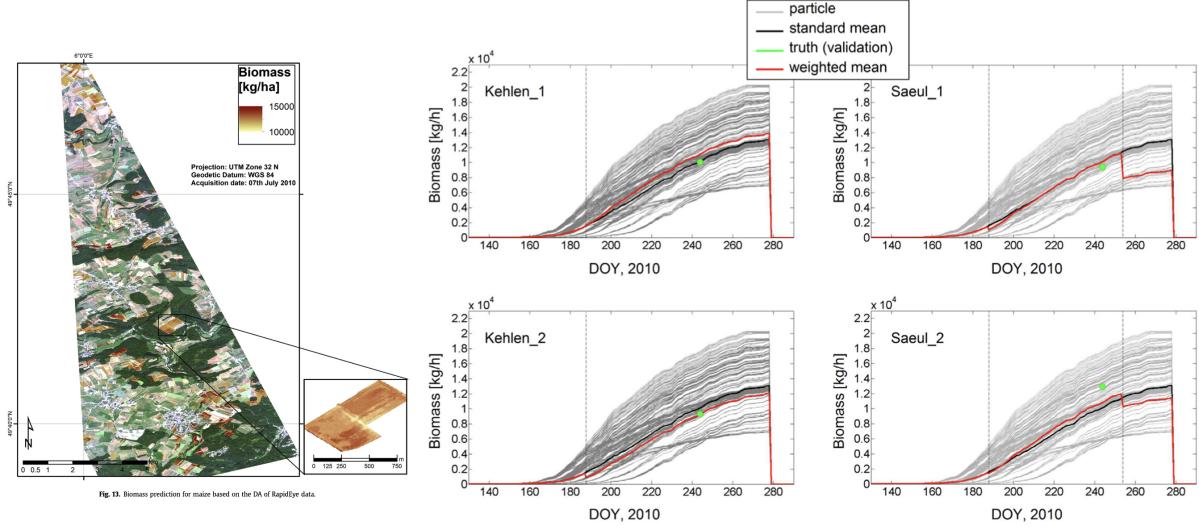


Figure 3: Collection of spectra data with RedEye (left) and the APSIM biomass predictive mean (black) as compared to the measured value (green) from simulations of various locations (right) from [11].

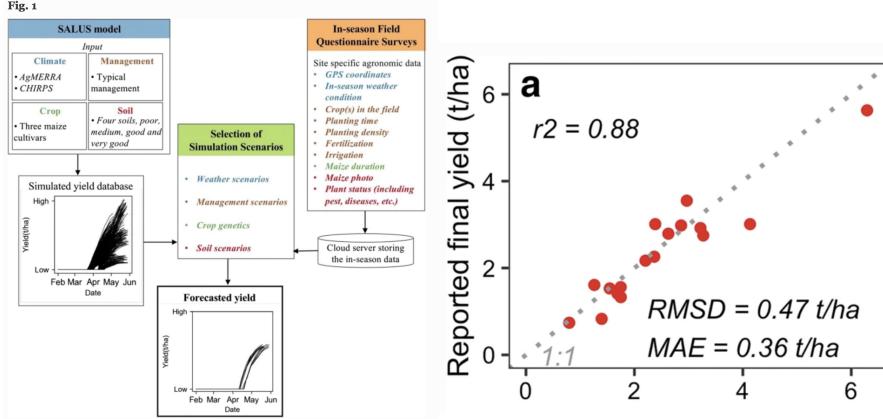


Figure 4: Process of field survey data collection and input to CGM (left) to improve biomass predictive accuracy with predicted vs. actual results (right) [9].

weather, soil, etc. data) and crop modeling output parameters (i.e., yield prediction, biomass prediction, water stress etc.) as input features for various machine learning models, including logistic regression, random forest, XGBoost etc. The authors were able to show a significant decrease in the root mean squared error (RMSE) for yield prediction and were able to isolate the most important input features for the ML models, with hydrological parameters being the most important [12]. Figure ?? shows the improvements in RMSE when coupling ML techniques with crop growth models for predicting biomass yield.

1.3 Our Project and Approach

With a variety of perspectives and approaches within the broad field of plant, crop and biomass modeling, our investigation seeks to develop a relatively simple crop growth model on a somewhat small scale. We set out to develop a virtual *greenhouse* to simulate a number of common vegetable, herb and/or house plants under varying environmental growing conditions. Given the time and resource constraints, our model is not intended to replicate the

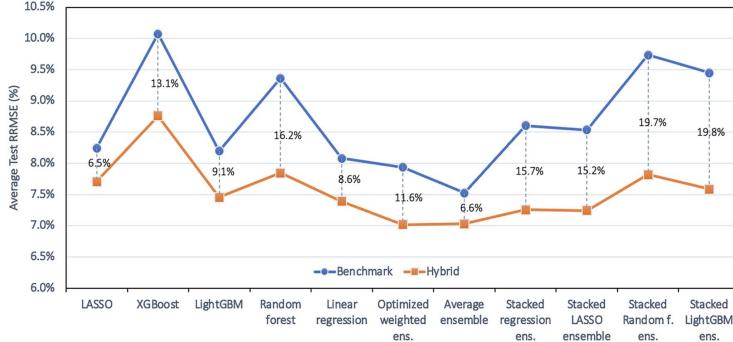


Figure 5: Improvements in root mean squared error rates of baseline and ML-APSIM crop yield models, demonstrating significant improvements for all methods, with an ensemble approach yielding greatest reduction in error [12].

level of complexity and advancement as has been discussed (above) and previously published for either individual plant structural-functional simulation or crop yield prediction. Rather, the aim is to develop a somewhat simple, yet predictive, emulation of a plant growth environment, accounting primarily for plant watering and sunlight uptake variables.

As such, the model developed here can allow for the creation of various virtual plant objects, which have their own representative properties such as color, growth rate, responsiveness to water and light. With this, virtual plants can be initialized as "seeds" either manually or randomly in positions throughout a Greenhouse area or "grid." By introducing simulated environmental factors such as water and sunlight at varying locations with gradating amounts throughout the Greenhouse area, the simulation can begin to predict rudimentary growth patterns of the plants, such as size, speed of growth. While the level of complexity for such a project can always be dramatically increased, i.e. to more accurately account for all environmental variables, plant-plant interactions and competition, soil properties, etc., the goal is to demonstrate a more proof of concept in a simplified, *fun* and relatively easy to understand form.

The focus of our model in this particular publication is to emulate and elucidate plant growth behaviour *particularly with a focus on differing plant watering patterns and differing plant species*. In each case, we look to monitor plant size (height, width, biomass) as well as water needed to grow and water acquired from the environment. Specifically, we look at four distinct watering patterns: even distribution, sequential grid watering, downhill watering and random watering (primarily focusing on the first three). We run several simulations with varying numbers of plants constructed, evaluating up to a total of three distinct plant "species" generated (each with their own color, water sensitivity and light sensitivity). With the focus on watering patterns, our model assumes equal light distribution throughout each grid point in the greenhouse, yet, plants that grow faster (and therefore higher) receive a greater amount of light relative to other nearby plants, which further stimulates faster growth.

2 Methods

To create the simplified crop growth model (CGM), a small python-based applet was created, represented at a high level in Figure 6. For the simulation, a number of presets and global variables are hard coded in the *run.py* file, including the greenhouse simulation size (height, length), number of timesteps to simulate, uniform water distributed, watering method as well as the number and "types" of plants to simulated. Plants are constructed using a Plant class, which specifies their location, appearance color, relative growth speed, water sensitivity (requirement) and status of growth (initiation, growing, wilting). *Run.py* constructs a Greenhouse from the Greenhouse class, which initializes the greenhouse properties including grid size, water drain rate, timestep and all associated functions to carry out the simulation/animation.

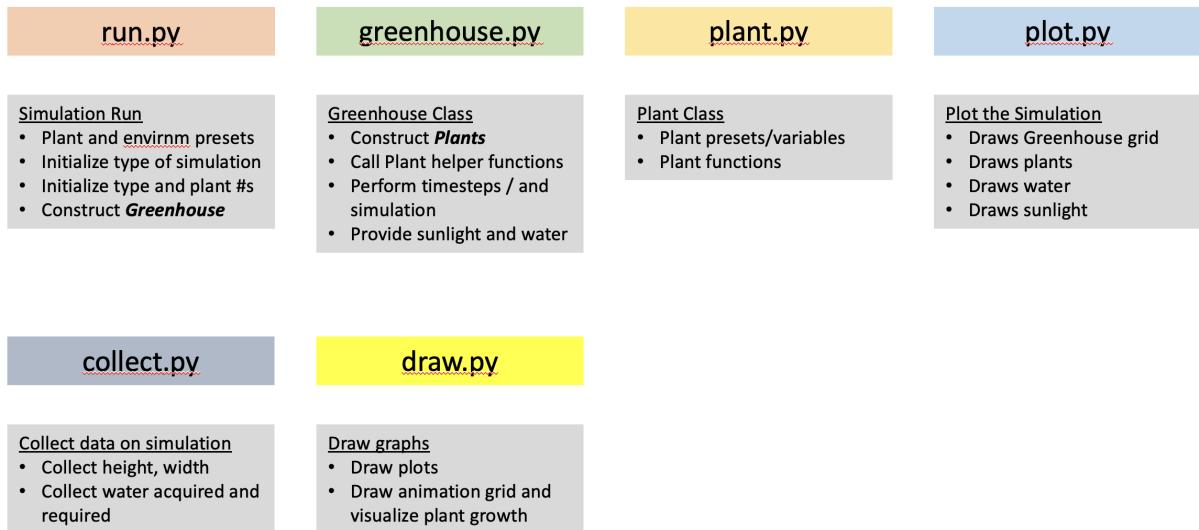


Figure 6: A high level view of the plant growth simulation applet

2.1 Key Simulation Functions and Parameters

The key functions for the simulation are largely contained within the Greenhouse class. While there are a number of helper and utility functions, the critical methods for the simulation of the Greenhouse and plant growth are:

- *add_plant*, which initializes the greenhouse grid with the plants;
- *timeframe_step*, which iterates through each frame of time, updates and calls the *water_plants* and *grow_plants* functions;
- *water_plants* function updates the greenhouse grid with water amounts from the type of watering method used (random or downhill);
- *uptake_water* function sends nearby water to each nearby plant. Higher water sensitive plants will not uptake as much water as more water "thirsty" plants (specified by the "w_effect" plant parameter);
- *add_sunlight* adds a sunlight particle to each nearby (defined as a set local distance) tallest plant. The sunlgith particle acts as a multiplier for the "l_effect" plant parameter, discussed below;
- *grow_plants* iterates through each plant and adds to each plant size (height and width) by multiplying the "w_effect" variable by the amount of nearby water ("water_amount")

In terms of how the simulation is created, each constructed plant is initialized with its own key parameters of *color*, *w_effect*, which controls the sensitivity to nearby water and thereby growth speed and *l_effect*, which controls how sensitive the plant is to uniform sun particles. The *w_effect* parameter and *l_effect* parameters collectively control how fast a plant will grow in the presence of nearby (required) water. The *w_effect* controls how rapidly the plant will uptake water from its surrounding environment. The amount of water taken up will subtract that amount nearby, limiting the amount available to surrounding plants as well as contribute to the amount a plant can grow. The *l_effect* acts as a multiplier for plants that grow higher faster. That is, in this model, light is considered equally distributed with each greenhouse grid point having its own and equal amount of light (one light particle per grid square) with tallest plants in a local minimum receiving the most light, acting as a multiplier to growth speed. This reinforces the growth rate of taller local plants.

Therefore, the relatively simple equation by which plant growth (height and width) is calculated each timeframe is shown below with the following variables:

- *light_amount* - plant class property that adds light amount if plant is tallest nearby plant. The wider the plant is, the more light will also be absorbed since light is absorbed by greenhouse grid squares plant encompasses.

- *water_amount* - plant class property that absorbs water based on the nearby water that has been distributed in the greenhouse grid.
- *w_effect* - plant preset acting as multiplier on growth given amount of water absorbed.
- *l_effect* - plant preset acting as a multiplier on the amount of sunlight absorbed. The amount of sunlight that is absorbed increases the max amount of water a given plant can absorb. The more water a plant can absorb, the faster it can grow.

$$\text{growth_width} = (w_{\text{effect}} * \text{plant}.water_{\text{amount}}) * (\text{light}_{\text{amount}} / \text{plant}.num_{\text{gridpoints}}) \quad (1)$$

$$\text{growth_height} = (w_{\text{effect}} * \text{plant}.water_{\text{amount}}) * (1 - (\text{light}_{\text{amount}} / \text{plant}.num_{\text{gridpoints}})) \quad (2)$$

As the plants increase in height, the tallest nearby plants absorb the greatest amount of sunlight, which increases the max amount of water a given plant can absorb. Once water is distributed to nearby grid points, the plants uptake the water, which is used to grow the plant in an x and y simulated direction. As shown in equation (1), the growth height depends on the *w_effect* preset, plant *water_amount* absorbed and the ratio of the *light_amount* to amount of plant gridpoints covered. The height depends on the inverse of this *light_amount* ratio.

2.2 Simulated Plant Type Sets

The simulation was tested using a variety of plant type sets that included combinations of tomato, lettuce, corn, basil and thyme plants (hypothetical plant types, each with their own color, light and water sensitivity). Characteristics of each plant species was adjusted to realistic parameters for testing. Each set was simulated under three watering procedures and factors of plant height, radius, water absorption, and water requirements were plotted using the simulation. Three sets 7 are presented here as examples of the model performance.

```
#set1
PLANT_TYPES = [((0.5, 0.39, 0), (0.001, 50), 'tomato'), ((0.5, 0.55, .0),
(0.11, 50), 'corn'), ((0, 0.6, 0), (0.13, 15), 'lettuce')]
#set2
PLANT_TYPES = [((0.5, 0.39, 0), (0.001, 25), 'tomato'), ((0.5, 0.55, .0),
(0.11, 10), 'corn'), ((0.49, 0.99, 0), (0.1, 25), 'basil')]
#set3
PLANT_TYPES = [((0, .39, 0), (0.13, 15), 'thyme'), ((.5, .55, .0), (0.11,
50), 'corn'), ((.49, .99, 0), (0.1, 25), 'basil')]
```

Figure 7: Sample of Plant Type Sets from Simulated Sets. Each plant has its own unique input parameters. In order, they are: color (rgb setting), *w_effect* param, *l_effect* param and plant name

3 Results

The results from four different simulations are presented below. Each simulation was ran with a uniform, sequential and downhill watering pattern. The sequential watering pattern distributes water stepwise in grid partitions with the downhill pattern distributing water more densely at the top of the greenhouse grid (simulating lower elevation). A simulation base case contains a single plant of each type from *set1* while the simulations for *set1*, *set2* and *set3* contain at least 50 plants per species.

When interpreting the following figures, the animation snapshots, shown in Figures 8, ??, ??, 9, 10 and 11 are of the final timepoint in the simulation, showing the final plant size. Water is represented as shades of blue, with darker hues representing a greater amount of water density. As is quite apparent, in many of the simulations we had instances where certain individual plants or plant species overgrew the greenhouse, covering the entire area. Each simulation was run for a total of 100 time frames.

3.1 Simulation Base Cases

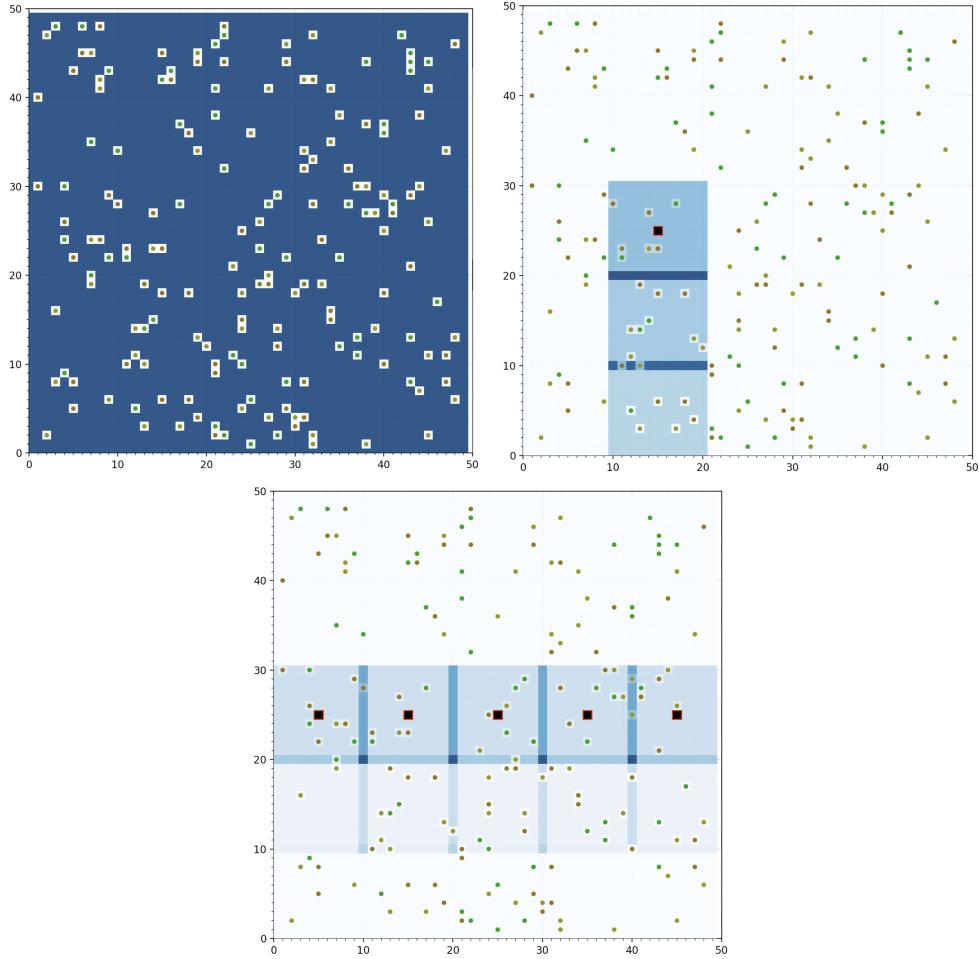


Figure 8: Set 1 Uniform|Sequential|Downhill Simulation

3.2 Simulations of Differing Watering Patterns

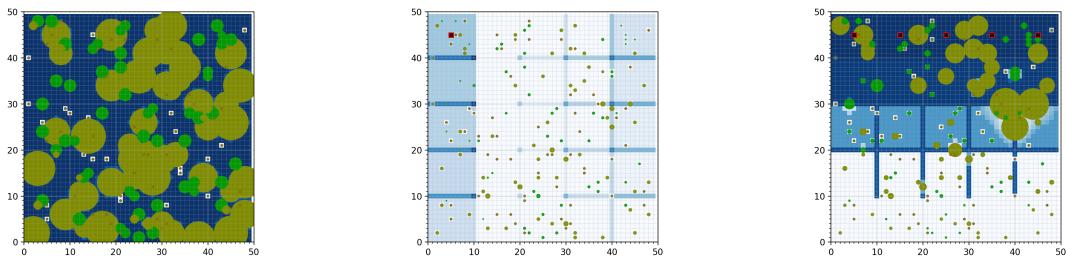


Figure 9: Set 1 Uniform|Sequential|Downhill Simulation

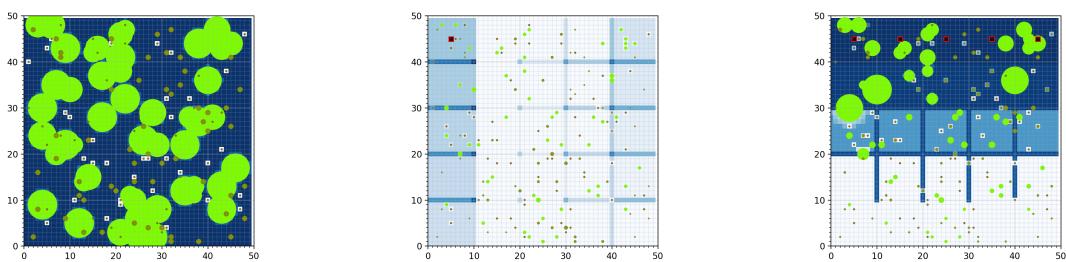


Figure 10: Set 2 Uniform|Sequential|Downhill Simulation

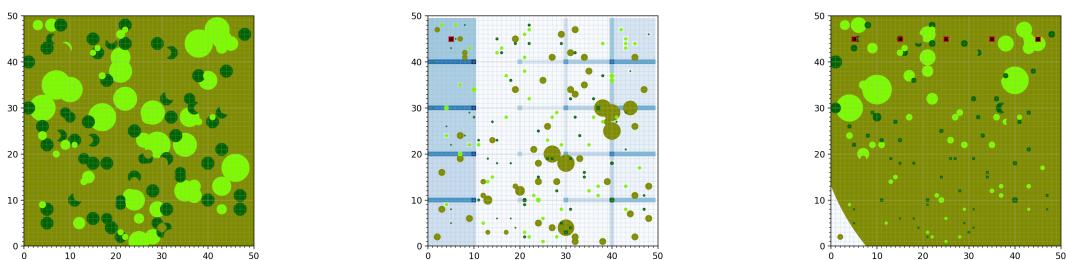


Figure 11: Set 3 Uniform|Sequential|Downhill Simulation

3.3 Set 1 Simulated Factors

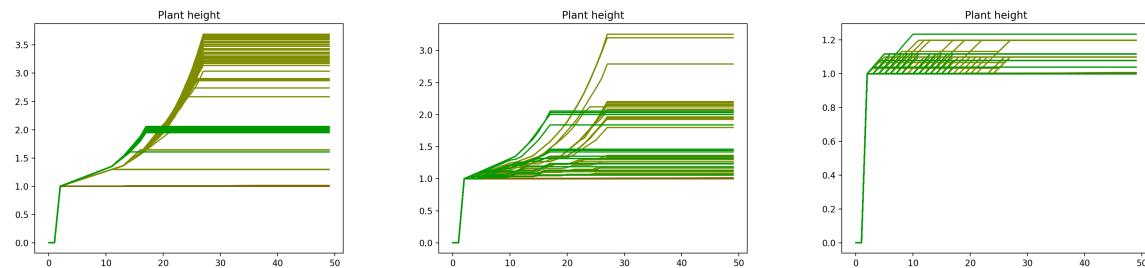


Figure 12: Set 1 Uniform|Sequential|Downhill Height

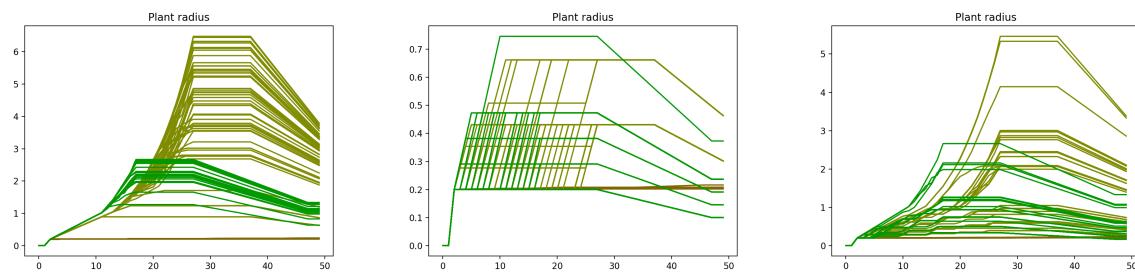


Figure 13: Set 1 Uniform|Sequential|Downhill Radius

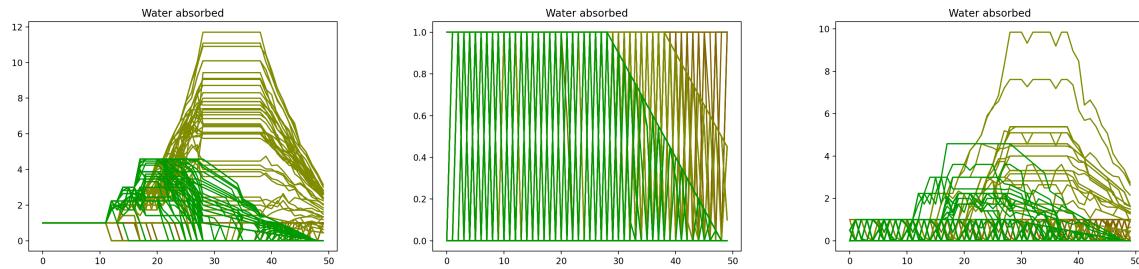


Figure 14: Set 1 Uniform|Sequential|Downhill Water Absorption

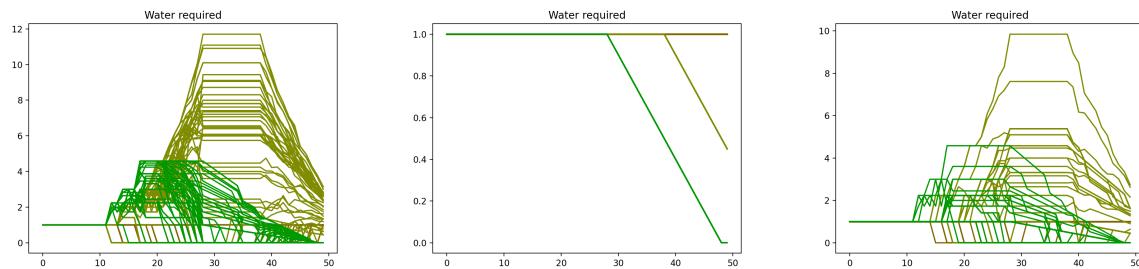


Figure 15: Set 1 Uniform|Sequential|Downhill Water Requirement

3.4 Set 2 Simulated Factors

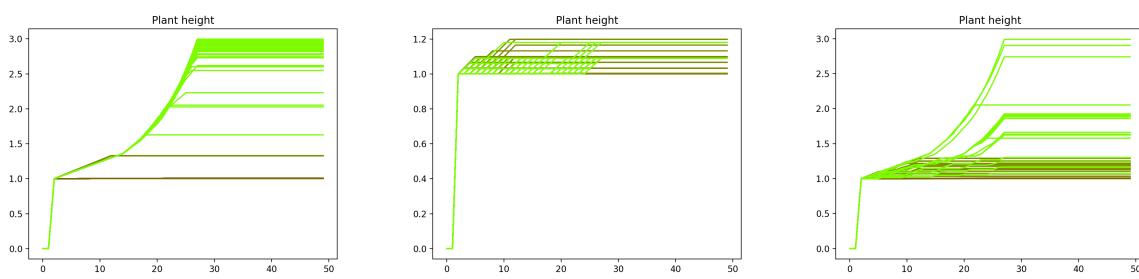


Figure 16: Set 2 Uniform|Sequential|Downhill Height

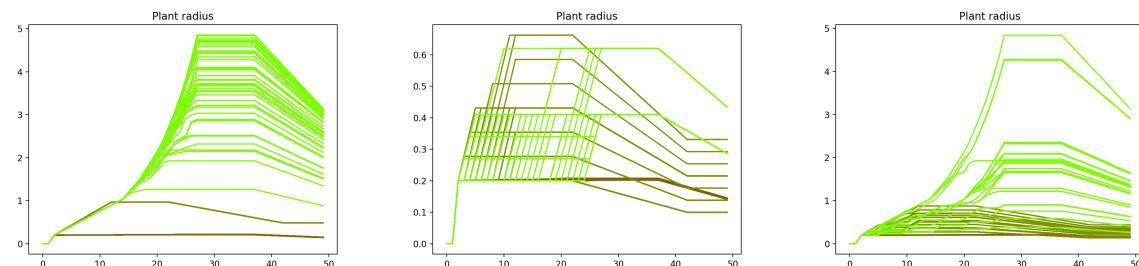


Figure 17: Set 2 Uniform|Sequential|Downhill Radius

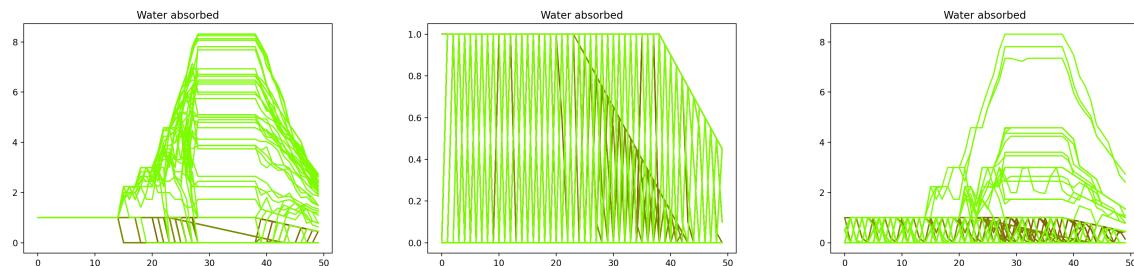


Figure 18: Set 2 Uniform|Sequential|Downhill Water Absorption

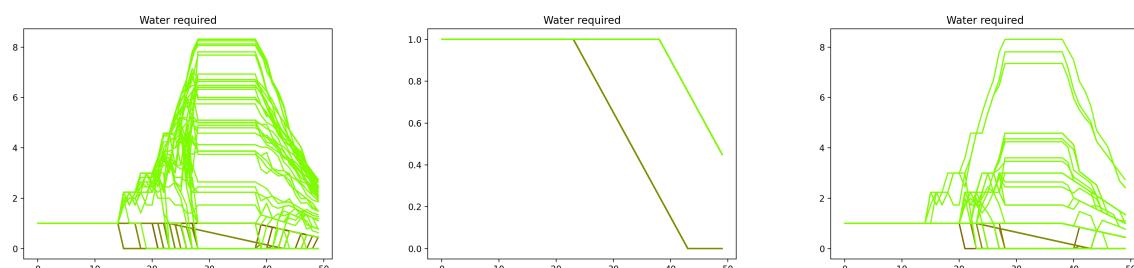


Figure 19: Set 2 Uniform|Sequential|Downhill Water Requirement

3.5 Set 3 Simulated Factors

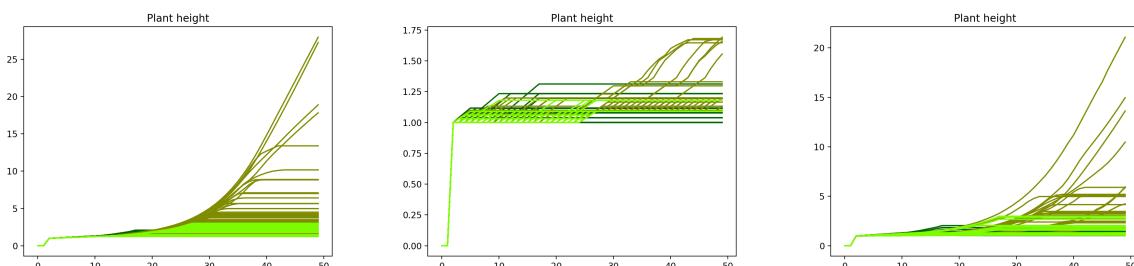


Figure 20: Set 3 Uniform|Sequential|Downhill Height

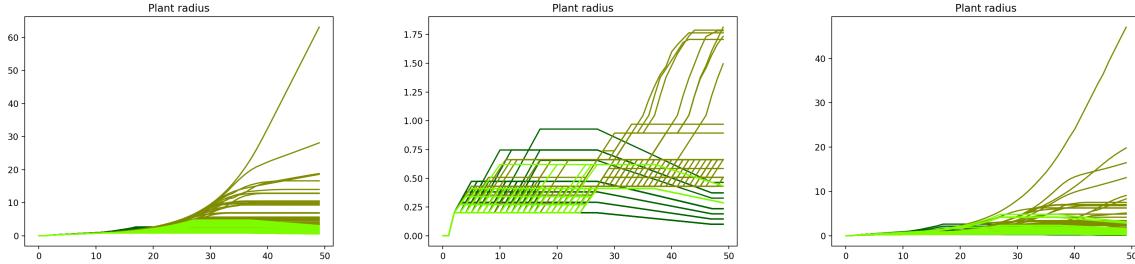


Figure 21: Set 3 Uniform|Sequential|Downhill Radius

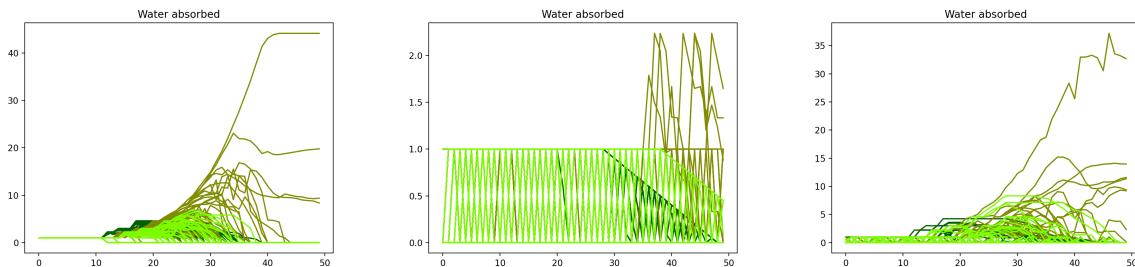


Figure 22: Set 3 Uniform|Sequential|Downhill Water Absorption

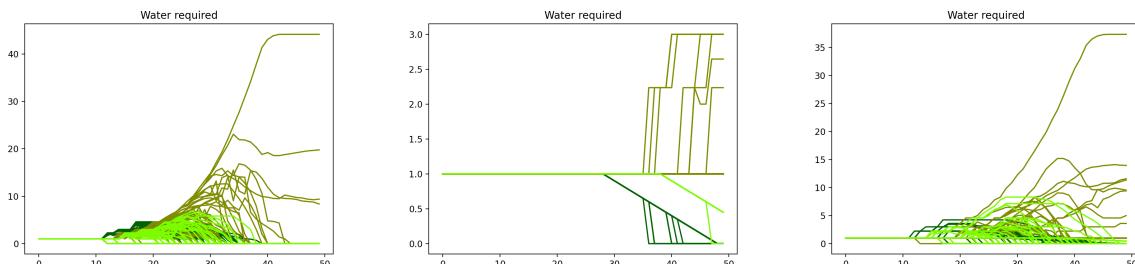


Figure 23: Set 1 Uniform|Sequential|Downhill Water Requirement

4 Discussion

Overall, the developed model did produce a somewhat realistic simulation of plant growth behavior in response to local water amounts and evenly distributed sunlight. By looking at three distinct watering patterns, we were able to evaluate 1) the effect simulated water has on plant growth and 2) whether the simulated behavior is close to biological plant growth.

Plants with an increased sensitivity to water and light (w_effect and l_effect) grew at much higher rates when provided with required water levels or placed in a runoff location. It is clearly observed from the animations and final plant states that plants within an area of high water density dramatically benefited and grew much faster.

The simulation also shows that all three species of plants overall grow better when downhill irrigation is used. Uniform irrigation of plants typically results in the overgrowth of a more dominant species. Sequential irrigation resulted in limited growth, with all species growing equally, but very slowly. Downhill irrigation, combining a uniform and

sequential irrigation system produced biomass growth that was most ideal for all three species. All species in set 1 were able to access the necessary resources for growth and competition was reduced when possible.

By watering plants downhill, the plants located at higher elevations grew at a decreased rate than those placed lower. Water runoff proved to be a very valuable resource to plants located downhill and greatly aided in the growth process of all species. Unlike set 1, set 3 showed an excessive overgrowth of one species when using downhill irrigation. This species, originally located downhill, was able to take over the majority of area sampled due to the excessive resources received. Plants originally located at higher elevations not only are deprived of water due to runoff, but also easily die off as a result of lack of nutrients and provide no competition to the overgrowing species.

5 Conclusion

The simulation proved capable of expecting impact of resources on plant growth and including considerations based on environment and other factors. Though only a few factors were testing in the early versions of this simulation, future work is capable of including a multitude of resources and growth factors. The inclusion of weather, sunshine, or human/animal impact on plant growth can aid in making the simulation a tool for many scientists, researchers, and even those simply planting a garden in their yard.

Considering differing environments such as forests, rain forests, desserts, parks, and cities allow for factors such as pollution, construction, climate change to be included. These factors mirror those of the real-world and allow for a realistic simulation that can help predict levels of $CO_2 : O$ production ratios between animals and plants.

Overall, the simulation has great potential to be an incredibly useful and efficient tool that can be implemented in many ways. Understanding our environment and working to better provide and care for it is a crucial part of humanities survival. Simulations, such as this, can be used not only for predicting plant growth in greenhouses or on farm land but to also predict the impact of human interactions on plant growth.

6 Contributions

References

- [1] Meyer et al. Stochasticity in plant cellular growth and patterning. *Plant Sci*, 5(420), 2014. doi:<https://doi.org/10.3389/fpls.2014.00420>.
- [2] Doak et al. Correctly estimating how environmental stochasticity influences fitness and population growth. *American Naturalist*, 166(1), 2005. doi:<https://doi.org/10.1086/430642>.
- [3] Kang et al. Analytical study of a stochastic plant growth model: Application to the greenlab model. *Mathematics and Computers in Simulation*, 78(1):57–75, 2008. doi:<https://doi.org/10.1016/j.matcom.2007.06.003>.
- [4] Lee et al. An automated, high-throughput plant phenotyping system using machine learning-based plant segmentation and image analysis. *PLOS ONE*, 2018. doi:<https://doi.org/10.1371/journal.pone.0196615>.
- [5] Jaeger et al. An automated, high-throughput plant phenotyping system using machine learning-based plant segmentation and image analysis. *J. Biosci*, 17(3):275–291, 1992. URL <https://link.springer.com/content/pdf/10.1007/BF02703154.pdf>.
- [6] Cournède et al. Greenlab: A dynamical model of plant growth for environmental applications. *ERCIM*, (61), 2005. URL https://www.ercim.eu/publication/Ercim_News/enw61/cournede.html.
- [7] Wang et al. A stochastic model of tree architecture and biomass partitioning: application to mongolian scots pines. *Annals of Botany*, 107:781–792, 2011. doi:10.1093/aob/mcq218.
- [8] Henke et al. Fspm-p: towards a general functional-structural plant model for robust and comprehensive model development. *Front. Comput. Sci.*, 10(6):1103–1117, 2016. doi:10.1007/s11704-015-4472-8.
- [9] Liu et al. Linking field survey with crop modeling to forecast maize yield in smallholder farmers' fields in tanzania. *Food Security*, 12:537–548, 2020. doi:<https://doi.org/10.1007/s12571-020-01020-3>.

-
- [10] Yang et al. Comparison and hybridization of two approaches for maize simulation. *Agronomy Horticulture – Faculty Publications*, 2003. URL <https://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1526&context=agronomyfacpub>.
 - [11] Machwitz et al. Enhanced biomass prediction by assimilating satellite data into a crop growth model. *Environmental Modelling Software*, 62:437–453, 2014. doi:<https://doi.org/10.1016/j.envsoft.2014.08.010>.
 - [12] Shahhosseini et al. Coupling machine learning and crop modeling improves crop yield prediction in the us corn belt. *Nature*, 11(1606):1–15, 2021. doi:<https://doi.org/10.1038/s41598-020-80820-1>.