Modeling Tree Growth and Resource Use with Applications

New Mexico
Supercomputing Challenge

Final Report
April 2, 2014

Team 58
La Cueva High School

Team Members:

Albert Zuo
Eli Echt-Wilson

Teacher:

Jason DeWitte

Table of Contents

Executive Summary

Previous work in tree modeling has generally either followed allometric rules for biological applications or focused on producing visually appealing trees. We propose and implement a unique model that simulates deciduous tree growth at the level of individual branches based on underlying biological processes. In our model, a tree is represented as a collection of finite-length segments. Leaves on the tree generate biomass for growth using photosynthesis. Light availability at each point in the canopy is calculated by propagating light through a voxel grid (3D index of space). The biomass is used for, in order, root growth, leaf replacement, segment growth in width, and growing new segments and leaves. Stresses on each segment are computed based on forces due to gravity and wind; growth in width is proportional to the stress. Branching is proportional to light availability at the branching site. Where possible, biological parameters are taken from naturally observed values and previous research. Extensive recursive algorithms and a voxel grid are used to achieve efficient simulation and 3D visualization; typical trees have 9,000 segments. We validate the model by confirming that generated trees exhibit allometric rules observed in nature, regarding leaf mass, stem mass, height, trunk diameter, and total mass. The allometric rules used to validate the model include: (a) In saplings, $M_L \propto M_T$ while in mature trees, $M_L \propto M_T^{0.75}$, (b) $H \propto D^{2/3}$, (c) $M_L = \alpha * M_S^{\beta}$. When exposed to varying light conditions, T-tests confirm that the model successfully predicts positive phototropism. When exposed to varying wind conditions (20 m/s and 30 m/s speeds), 2-sample T-tests confirm that the model successfully predicts anemotropism. Our model exhibits these emergent properties without explicitly programming them into the model. The model and software has potential applications in forest management including the maximization of carbon sequestration and timber production.

1. Introduction

In nature, trees follow sets of allometric rules, but it is not known why trees follow these rules. Allometric rules are relationships between various quantitative values in trees that hold true for all tree growth. For example, across a broad range of tree species, the height of a tree is roughly proportional to the diameter of its trunk raised to the ⅔ power. However, it is not known exactly how varying environmental conditions affect tree growth. Creating a tree model that is based on accurate biological principles would not only provide a possible explanation for these allometric rules, but would also be useful in predicting if and how trees will grow in a given environment.

Expanding the single model to a forest model is something that would be useful in environmental applications. For example, tropical tree plantations are known to sequester carbon at 20 to 30 times the rate of old-growth forests [D13]. However, these tree plantations provide very different growing conditions than in nature, so the allometric rules cannot accurately predict carbon uptake. Experiments to optimize tree plantation growing conditions take decades [TJT+01]. An accurate computational model of tree growth and interactions between each other could allow much faster experiments to optimize factors such as carbon dioxide uptake and land space management.

We propose and implement a unique model that simulates deciduous tree growth at the level of individual branches based on underlying biological processes. Leaves on the tree generate biomass from photosynthesis, taking into account shadows. The biomass is used to thicken existing branches based on physical stresses they experience, as well as to grow new branches and leaves. Powerful computing techniques including extensive recursive algorithms and a voxel grid (3D index of space) are used to achieve efficient simulation and 3D visualization. We validate the model by confirming that generated trees exhibit allometric rules observed in nature, regarding leaf mass, stem mass, height, trunk diameter, and total mass, as well as phototropism and anemotropism.

An accurate model of tree growth allows the ability to test tree growth for many emergent properties: tree interaction, growing into light spaces, growth in response to building or other shadows, etc. We plan to make the source code available on an open-source license, so that other groups can use and extend it for additional simulations.

2. Background and Related Work

Tree modeling has traditionally been undertaken from two different perspectives: modeling based on graphics and modeling based on biological accuracy (see [TZW+07] for an image-based model and [HN09] for a rule-based model). However, due to factors including but not limited to geometric complexity, ease of modeling and editing, and realism, scientists have not created a successful model that combines the two perspectives.

Rule-based modeling has traditionally relied on programming specific aspects of tree growth into the model, and using the model to observe other characteristics. For example, Sean Hammond and Karl Niklas explicitly program allometric rules into the model. Trees are represented as a single trunk segment with a hemispherical canopy. Trees in the model grow based on light availability for a tree. The model is used by modeling thousands of trees in a forest, and observing how competition for light affects population dynamics. While this model is very useful, we hope to extend it by increasing the accuracy of specific trees and modeling growth by accurate biological processes such as photosynthesis and growth apportioning. If our proposed model is indeed accurate, it will exhibit the allometric rules that are explicitly used by Hammond and Niklas [HN09].

Image-based modeling is very useful and popular in the movie and gaming industries, where it is useful to generate visually appealing trees [e.g. TZW+07]. From a biological standpoint, however, these models are not based on any rules or principles. The models focus on the use of L-systems and advanced computing techniques to simply reproduce images of real trees. The models are not useful for simulating tree growth over the lifetime of a tree. Although our approach focuses on modeling individual branches of the tree and therefore has a graphics output with visual appeal, the aim of our project is to create a biologically accurate model rather than a model that could be used for gaming or video industries.

To our knowledge, there is no work in tree modeling that achieves accurate simulation based on local environmental factors and inherent biological processes. Using advanced computing techniques, we have created a biological model of tree growth at a level of individual branches. The goal of our model is not only to bridge the gap between image-based and rule-based models, but to create a unique mathematical and biological model of tree growth that exhibits emergent properties of growth without explicitly programming them into the code.

3. Mathematical Model

Trees grow based on a series of biological and physical processes. At a high level, the tree uses the process of photosynthesis to fix carbon from carbon dioxide into carbohydrates for growth and cellular respiration. The carbohydrates generated from photosynthesis are apportioned across the tree for growth.

Our model focuses on the scarcity of light as the limiting factor in the process of photosynthesis. Each step of our model, in which all values are calculated and represented, is equal to a time period of one year. Available light in the tree canopy is determined using methods of light propagation. We propose that in the step where carbon must be apportioned across the tree for growth, carbon is first devoted to growth in width based on the stress experienced by a branch, and the leftover carbon is used to branch and grow new leaves. Our model represents trunks and branches of the tree as *segments* of finite length, with branching occurring only at the ends of segments.

3.1 Available Light

In order to calculate the solar radiation incident on a canopy over a period of a year, it is necessary to integrate the incident radiation as the sun moves from morning to evening and through all of its positions in the sky over a year. Some light is incident as direct beam radiation, while other light is reflected off clouds and other objects, arriving at the tree as diffuse radiation. The amount of each available each year depends mostly on the latitude of the tree and the typical cloud cover. Since the two-variable integration needed to calculate this is rather complex [N05], we opted not to perform these calculations and instead chose to consult an online source to obtain the solar irradiance per year [SEH13]. Our model approximates light to travel in 5 directions: straight down, plus light coming diagonally from North, South, East, and West.

Importantly, the same amount of light is not present at all points in and under the tree canopy. This is a critical factor to model tree growth accurately. To determine the amount of available light at given points on the tree, light must be propagated through the canopy, where if leaves are present in the space the light is going through, a certain amount is absorbed by the leaves (see below). This results in a smaller amount of branches in the middle of the tree, since branches in the middle are shaded by leaves at the top of the canopy. The shadow is present at all points inside and under the tree canopy; at the bottom of the tree, the shadow can be seen on the ground based on the amount of light left.

3.2 Photosynthesis

Plants add dry mass almost exclusively by photosynthesis, which creates chemical energy in the form of carbohydrates from light energy. Chloroplasts within the cells of trees take in light, water, and CO_2, and through a series of chemical reactions, make a monosaccharide. This monosaccharide is used to form more complex carbohydrates which are used for growth. Photosynthesis is a complex process involving many chemical reactions catalyzed by different enzymes. However, the process can be summarized by the general reaction:

$$6CO_2 + 6H_2O + \text{light} \rightarrow C_6H_{12}O_6 + 6O_2$$

where $C_6H_{12}O_6$ represents a carbohydrate such as sucrose or glucose. These carbohydrates can be assembled into long chain sugar molecules like starch and cellulose, which is the main portion of cell walls. The CO_2 comes from the air through the leaves' stomata. H_2O is mostly absorbed from the soil, and light comes from the sun.

All three of the input quantities can be a limiting factor for the process of photosynthesis, although in practice CO_2 is normally abundant. In desert environments, water is usually the limiting factor, but in other locations such as the tropics or many American forests, the availability of light is the limiting factor. Other additional factors that affect the rate of photosynthesis include the leaf temperature and the rate of gas exchange through stomata, but those are rarely dominant factors [CN98, ch. 14]. (See [CBGB91] for a more complex model, as well as [CN98, Sec. 14.8].) In this project we assume that water and CO_2 are abundant, and we focus on the influence of light availability on plant growth.

Monteith observed that biomass accumulation by plants is proportional to the solar radiation collected by the plants. This suggests a model

$$A_{n,canopy} = ef_S S_t$$

where $A_{n,canopy}$ is the net biomass accumulation of the plant canopy in g/m^2/day, S_t is the total solar radiation incident on the canopy in MJ/day, f_S is the fraction of incident solar radiation intercepted by the canopy, and e is the conversion efficiency for the canopy [Mon77, CN98]. e must express the fraction of solar radiation that the plant can use in photosynthesis (photosynthetically active radiation or PAR) as well as the fraction that is absorbed or reflected. Monteith reported e values around 1.5 g/MJ for a number of plant species. More recent work limits the solar radiation to the PAR portion, and expresses e in molar units such as (mol CO_2) / (mol photons). Absorptivity of individual leaves is about 0.5 (0.85 of PAR), but once the leaves are arranged into a canopy, internal reflections among leaves raise absorptivity of the canopy to 0.85 (0.95-0.98 of PAR) [CN98]. In our model, units will be expressed in kg/m^2/year.

The net biomass accumulation calculated as shown above is the amount of dry mass that can be apportioned to growth throughout the tree. As discussed in the next section, some of this mass will be used to grow woody tissue for branches, roots, and the replacement of old leaves, while the remaining mass will be used to grow new branches and leaves.

3.3 Growth

A tree uses glucose and other products of photosynthesis in order to grow. One of the main questions to be determined is how the tree apportions growth among its various parts. While a tree does have systems in place to carry out paracrine, or local signaling, it does not have an endocrine system to perform long distance signaling. This means that cells in different regions of the tree do not have communication to determine how much of the available glucose they should use to grow. However, trees achieve a set of allometric rules relating sizes/masses of various parts. If a tree has no way of calculating these masses and communicating it throughout the organism, the question remains as to how it distributes its mass.

We propose one possible method by which trees could apportion their mass growth throughout the tree. In our model, the growth of each segment is linearly proportional to the mechanical stress that the segment is under from gravitational and environmental forces. When under mechanical stress, trees release the hormone auxin which stimulates the release of cambium. Cambium is directly involved in the stimulation of secondary growth, which is the thickening of shoots and roots [ASG+11].

The two main stresses that a tree is under are from the gravitational force of its branches and leaves and the forces caused by wind blowing through the leaves of the tree.

The gravitational force on a segment is easy to compute. It follows

$$F = mg$$

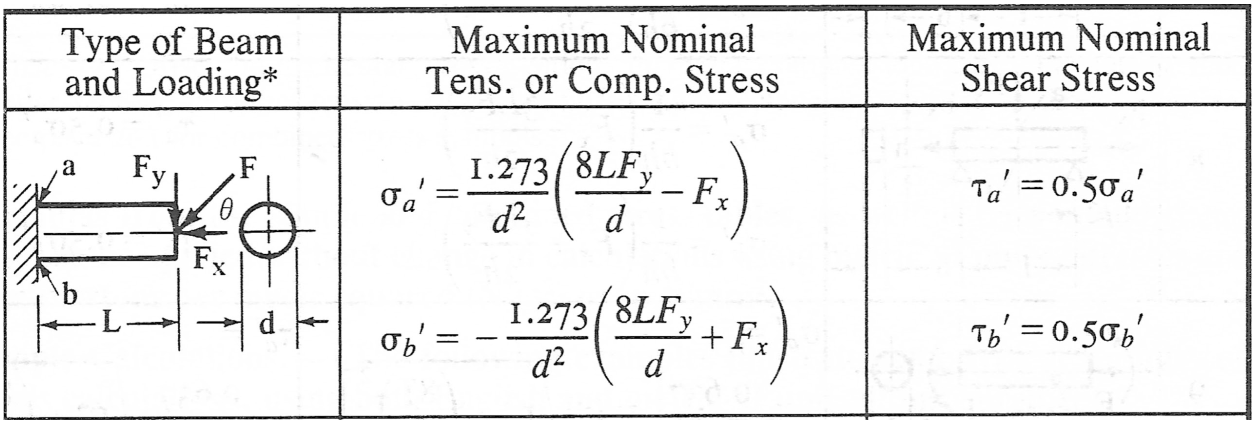
where m is the mass of the segment and its descendants (in kg) and g is the acceleration due to gravity (9.8 m/s^2). The gravitational force is applied at the center of mass of the descendants of the segment. Note that the forces on segments near the leaves of the tree will be small because there is little mass and it is close to the segment, while the forces on major limbs and the trunk will be significantly larger and have a long moment arm.

The wind force on a segment is computed by finding the magnitude and center of the wind force felt by its descendants. The force experienced by a single leaf from wind is:

$$F = 0.5c_d rAs^2$$

where F is in Newtons, c_d is the drag coefficient of the leaves, r is the air density (in kg/m^3), A is the area of the leaf (in m^2), and s is the wind speed in m/s [BW06, N05]. The drag coefficient c_d may vary by leaf shape [BW06, V09] and the maximum horizontal wind speed can vary by location, so both are adjustable parameters in our model. After computing the center of the wind force of all leaves that are descendants of a given branch, the x and y components of the wind force on a given segment are converted into the perpendicular and parallel components, respectively, of the force on the segment.

Our model makes the simplifying assumption that all branches (segments) are cylindrical. The equation for the stress experienced by a cylindrical beam from an attached mass is [MH25, p. 207]:



Type of Beam and Loading*	Maximum Nominal Tens. or Comp. Stress	Maximum Nominal Shear Stress
	$\sigma_a' = \frac{1.273}{d^2}\left(\frac{8LF_y}{d} - F_x\right)$ $\sigma_b' = -\frac{1.273}{d^2}\left(\frac{8LF_y}{d} + F_x\right)$	$\tau_a' = 0.5\sigma_a'$ $\tau_b' = 0.5\sigma_b'$

a
F_y
F
θ
F_x
b
L
d

where F_x and F_y are the parallel and perpendicular components of the applied force, L is either the distance from the base of the segment to the center of mass of the segment and its descendants or the distance from the base of the segment to the center of the wind force on the segment and its descendants, and d is the diameter of the segment. The maximum stresses from wind and gravity are summed [MH25].

The first equation, in context, calculates the maximum tensile stress experienced by the branch, which occurs at point *a*. The second equation calculates the maximum compressive stress experienced by the branch, which occurs at point *b*. Since wood is much more vulnerable to compressive force than tensile force [MH25, p. 375][N09, p. 336], our model only calculates the compressive force when determining its growth.

After calculating the stress it experiences from the equations above, the tree grows in width proportional to the stress experienced. The constant of proportionality is a parameter of the model and can depend on the tree species. The tree will constantly add width, and the amount of growth in width will increase if more is needed to handle the stresses from gravity or wind. If the tree loses weight from the death of branches and therefore experiences less stress on the trunk and branches, then the growth in width will be significantly smaller.

Trees are largely composed of dead tissue. The heartwood of the tree does not need to be replaced each year, and so when calculating the carbon used, only the addition in width is calculated; replacement of previous wood is not needed.

3.4 Branching

After the tree has apportioned the required mass to handle these stresses, the leftover carbohydrates created from photosynthesis can be used to grow new branches and leaves for the following season.

In our model, the chance of branching at any location in the tree is proportional to the light available at that location. This makes sense because a tree should invest carbon in areas where it will be able to grow leaves and have available light to carry out photosynthesis. Branching in nature is more common where light is present [FKKC10][EKVS11]. When the tree branches, a new cylindrical segment is created at the end of the parent segment, creating continuous woody tissues. The dry mass (biomass) of the new segment must be subtracted from the total carbon available for growth.

Our model assumes that leaves are present on the youngest tissues of the tree. Therefore, branching will create new tissues that have leaves on them, and is critical to the survival of the tree. If the tree has no leftover carbon to devote to branching and therefore growing new leaves, then the leaves will fall off over the winter and will no longer generate carbon for growth. Therefore, a tree that is not able to branch and grow new leaves will die.

3.5 Death of Branches

Trees stop apportioning carbon to branches that do not produce any carbon. Those branches will not have new material, and can be considered dead since they are only composed of heartwood. Within our model, segments that have no leaves die (no carbon produced). Since leaves are a direct result of branching, segments that have a low amount of light have a higher chance of dying. This local principle will also be true for the tree as a whole. When the tree no longer produces carbon, and has no leaves at the end of the growing season, the tree is considered dead.

4. Implementation

We implemented the biological model of tree growth in a program called 3DTree. As noted above, in our model a *segment* is the basic building block of a tree, representing a piece of trunk or branch between branching points. In each time step of one year, the program:

1. computes the light on the leaves and the mass of carbohydrates created through photosynthesis
2. uses some of the initial carbon to grow roots and replace leaves lost over the winter
3. computes stresses on all segments from gravity and wind
4. grows segments in width corresponding to the stresses they experience
5. uses leftover carbon to grow new branches and leaves the next year.
6. prunes branches that are not harvesting carbon

To accomplish this simulation efficiently, the tree is represented as a recursive data structure, with a Voxel Grid as a spatial index of the area it grows in. These data structures are described in more detail below.

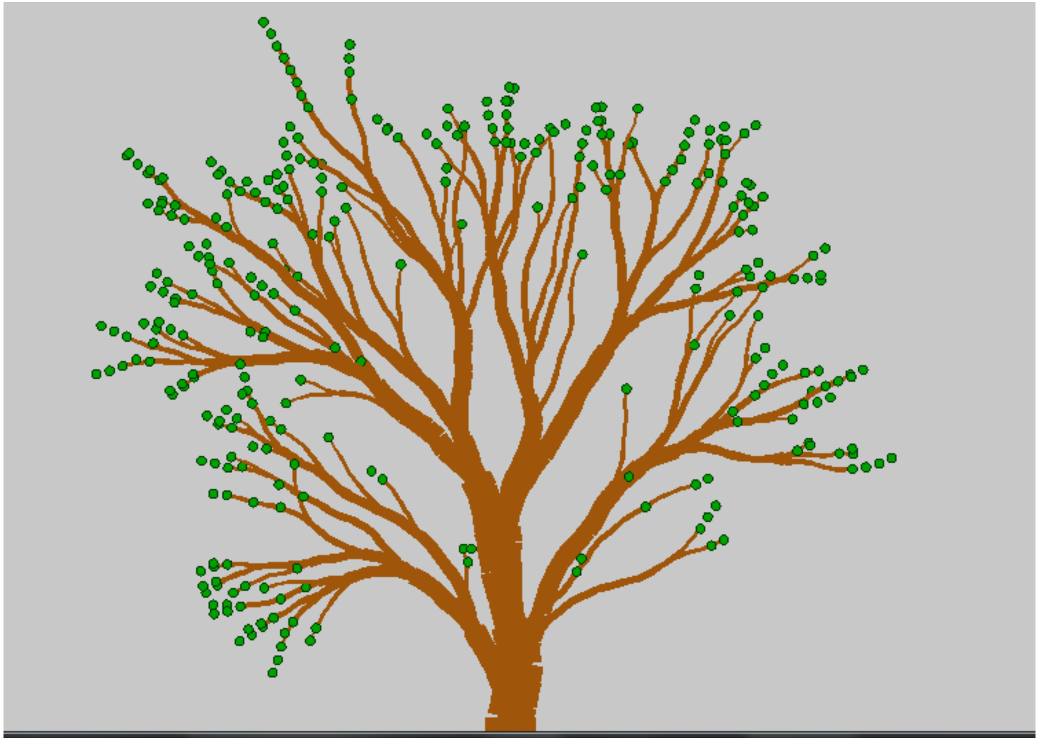
3DTree was implemented using Java and the Eclipse programming environment on Mac and Windows computers. Graphics for visualizing the tree structure, growth, and shadows were implemented in OpenGL. The full code listing is linked in Appendix E.

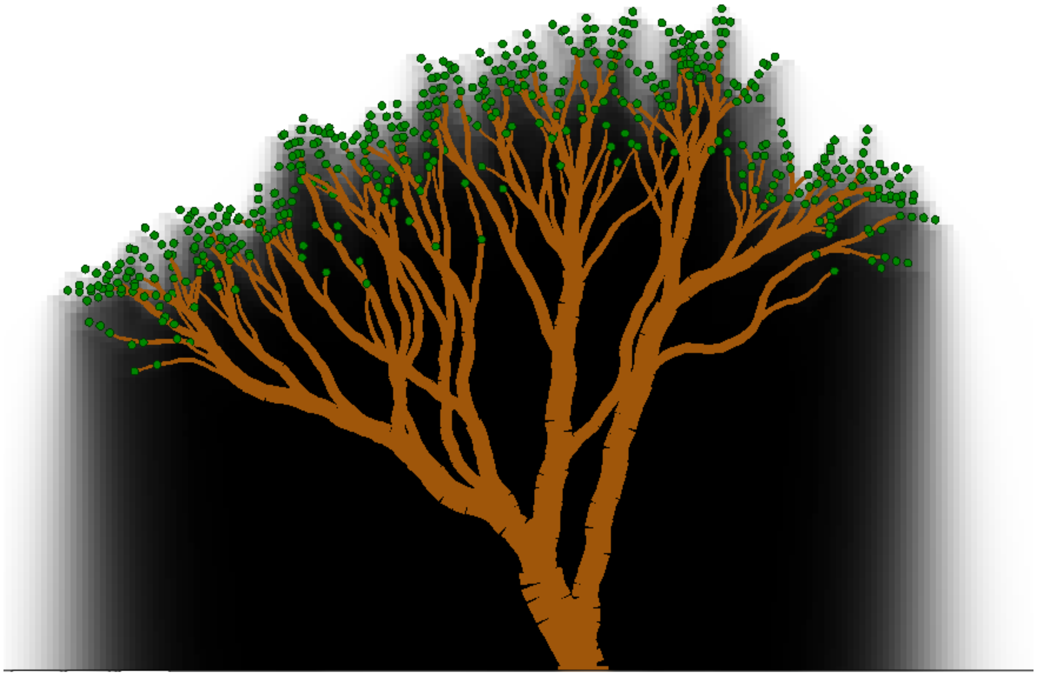
4.1 2-D Tree Implementation

As a stepping stone to our final 3-D model of tree growth, we implemented a basic 2-D model. Like the 3-D model, the 2-D model branches based on light availability and has a 2 dimensional voxel grid to store light and space. However, it does not carry out photosynthesis, and the width of each segment is arbitrarily chosen to be proportional to the square root of the number of descendants it has.

While the 2-D model was a very important part of our implementation because it allowed us to familiarize ourselves with the concepts of a voxel grid and tree branching, we eventually dropped the 2-D model in favor of a 3-D one. The limitations of two dimensions prevented us from implementing many important parts of our biological model, and wasn't capable of producing the same accuracy that the 3-D model offered. For example, volume and mass have essentially no meaning in 2 dimensions, and cannot be computed. As a result, our mathematical model that relies heavily on a carbon budget (in units of mass) and stress (computed as a function of mass) could not be represented in 2-D.

The figures below contain examples of trees created in the 2-D environment. The first figure shows a simple tree before we implemented light and shadow propagation. The second figure shows a tree with vertical light propagation; 2-D allows one to visualize the effects of shadows inside the tree that also affects the 3-D simulation. In particular the shadow causes there to be fewer leaves under the canopy and affects the overall shape of the tree.





4.2 3-D Tree Implementation

A tree in our model is a collection of segments, or localized clumps of cells. A segment is a cylinder with a constant height but a diameter that grows over time. A tree begins as a single root segment, and the root branches upward to create additional segments. Each segment has a parent segment that it sprouted from and a list of children segments that directly sprout from the segment. Our model treats every segment as a wood segment, and only the newly created segments have leaves present on them. A segment also has a function to get all of its descendants, which includes every single segment that is either a child segment, a child of a child, etc. In order to update every segment of the tree for each tick, which represents one year, the program must update the root and all of its descendants. This is accomplished through a set of recursive algorithms.

Trees are naturally represented as a recursive data structure, due to the fact that the trunk has branches, which have more branches, which have more branches, etc. until leaves are reached. Therefore, in order to create a model of a tree, recursion must be viewed as a critical component to master. Recursion is the repeated application of a procedure or structure until a finite base case is reached. In our case, recursion consists of carrying out a procedure on a given segment, and the procedure includes calling itself on the segment's children. When it acts on the children, it will then be called on the children's children, and so on. At the point where a segment no longer has any children, that is called the base case, and it no longer needs to recurse. Our program uses recursive algorithms in many different ways. For example, to calculate the total mass of a segment and all its descendants, we recursively cycle through its children segments, adding the mass of the segment each time. The mass of a tree is computed by applying this procedure to the root segment.

In each tick (year) of the simulation, the computer must first calculate the total carbon budget available for growth. To do so, it recursively goes through the segments, and, if a leaf is present on the segment, it calculates the amount of carbon generated by the leaf based on the equations given in our mathematical model. The total carbon budget is the sum of the carbon generated from each leaf, which is then used for growth. Since our model accounts for above ground mass only, a fraction of the carbon budget that would be used for roots is subtracted from the total. In addition, since a tree loses its leaves every year, we subtract the current dry mass of the leaves from the carbon budget, so that the tree will be capable of regrowing these leaves in the new season.

As described in the mathematical model, carbon is next used to grow segments in width proportional to the stress that they experience from wind and gravitational forces. Once again, recursion is used to cycle through all of the segments and calculate the width of the segment based on the stress, which is calculated using equations presented in the mathematical model. The width of every segment on the tree increases each year proportional to the stresses felt by the segment. The added mass is subtracted from the carbon budget.

After a tree has grown in width to handle the stress from environmental forces, leftover carbon can be used to branch and create additional segments. This step, however, is not done using recursion. Due to the nature of recursion, entire branches would be able to use the leftover carbon before others, leaving no carbon for other branches. In order to avoid this issue, we use recursion to get a list of all of the segments in the tree, but sort the list by the ages of the segments. With the sorted list of segments, we can then update the branching and growth of the tree more uniformly and equally across the tree by branching first on young branches.

When a segment is called to update branching, its chance of branching is proportional to the light available at the segment, computed using the Voxel Grid (below). When the tree branches, a new segment is created with a direction that could be somewhat different from its parent segment direction based on a randomly generated 3-D vector; the maximum angle is a parameter of the tree species. The start of the new segment is in the same place as the end of the parent segment, creating contiguous woody tissues.

When each segment is updated, it is checked to make sure that it is still living. If a branch has not branched and created new segments for 3 straight years, then there will be no leaves present on the branch. When this is the case, the branch (any segment without a descendant that has leaves on it) will die and will be removed from the simulation. This is a way for the tree to lose branches that are no longer growing and generating useful carbon for growth.

Constants for the growth of the tree, such as the branch angle factor, photosynthesis efficiency constant, and mass of a single leaf are stored in a separate class called TreeParams. This allows the code to be more well organized, and means that when we desire to change certain variables we can simply go to TreeParams and fix it in one place instead of searching through the code to find where we used that variable. The code for the TreeParams class can be viewed in Appendix F, which contains the input parameters that we used for a deciduous tree.

4.3 Voxel Grid

A critical component of the 3-D simulation is the implementation of a voxel grid. A voxel grid is a 3-dimensional index of space with constant spacing of voxels. Each voxel ("voxel" is a combination of "pixel of volume") represents certain information about the cube of 3-dimensional space that it covers, such as whether it is occupied by segments and/or leaves, the light coming into and out of the voxel, and so on. Voxel grids are typically used in 3-D computations and visualization; examples include 3D scene reconstruction from multi-view imagery (e.g. [NZSS13]) as well as the computer game *Minecraft* [M09].

The voxel grid is critical for our tree simulation's efficiency. It allows us to efficiently store information such as the available light entering a cell, the light leaving a cell, and whether a cell is occupied with a branch or leaf. When segments are added to the tree, the tree can quickly find the location of the segment in the voxel grid; it sets the location to be full. When another segment tries to grow into this occupied space, it is unable to. In addition, our model incorporates a wrap-around world so that there are no edge effects. For example, a branch that goes off the east side of the simulation will enter on the west side. This feature allows us to essentially model a small plot within an infinite forest, which is useful for our application to maximize carbon sequestration in a large tree plantation.

A critical function of the voxel grid is light propagation through the tree canopy. Our model is capable of propagating light in two different ways: directional and non-directional. Non-directional propagation is a more uniform way of propagating the light in which the direction of light rays are not stored. The light going into any given cell is a function of the light exiting from five of the cells above it: the cell directly above it, where it gets most of the light, and the four cells that are above it and to the north, south, east, or west, where it gets a small fraction of the light. When a cell is occupied by a branch or leaf, the light leaving the cell will be significantly smaller than the light that entered the cell. Since this form of light propagation will not store the direction of the light rays, it results in a very uniform shadow without holes where the light made it through gaps in the branches. Figure 4.3.1 shows an example of a shadow with non-directional propagation:

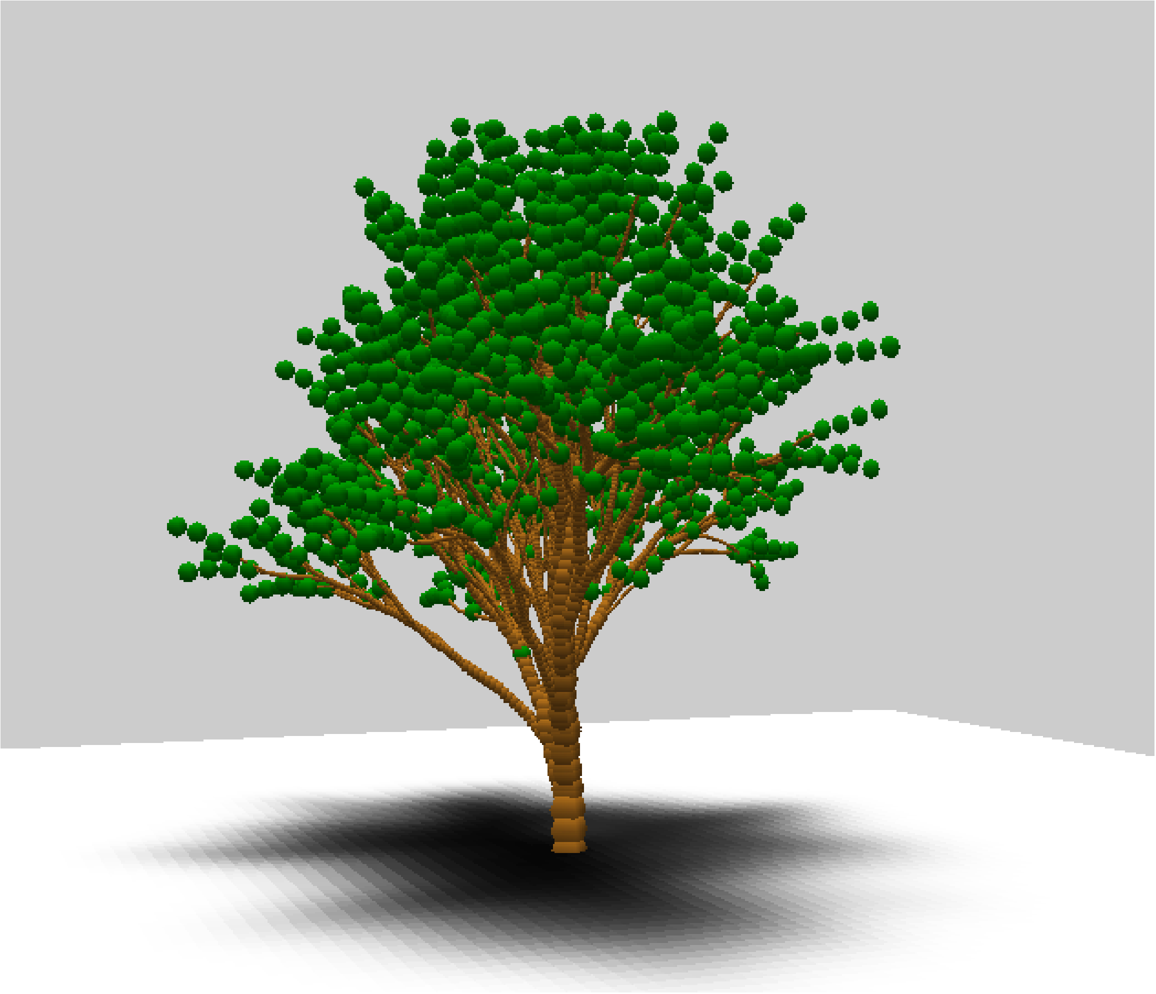


Figure 4.3.1 Tree with non-directional light propagation

Directional propagation is more representative of real light. In directional propagation, we approximate the many directions of light propagation with 5 directions. Each cell has an array of light from multiple directions: straight down, north, south, east, and west. The total light in a cell is the sum of the light from all the directions, but the light that exits a cell is directly sent to the cell opposite the cell the light came from respective to the cell. Light in the down direction will be propagated directly downward through cells, and light from the northern direction will propagate diagonally downward and southward. Light exiting any cell that is occupied by tissues of the tree is significantly reduced, because it is absorbed. The light could be approximated by additional directions, but the main issues of light availability and shadowing in the tree canopy are relatively well represented by the 5 approximate directions. The directional form of light propagation is more accurate than non-directional, and is therefore the form of light propagation that we use in our final model. Figure 4.3.2 shows an example of a shadow on the ground with directional propagation.

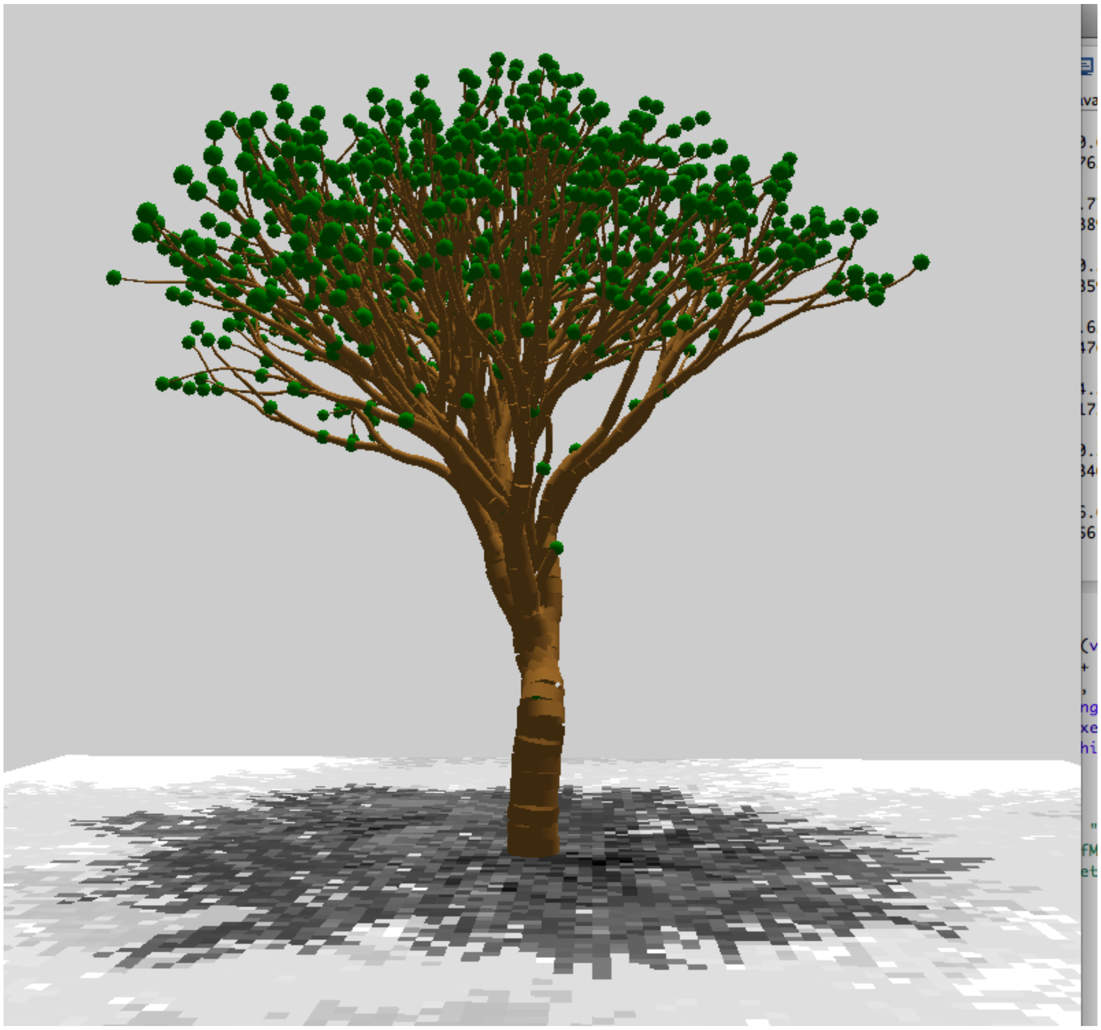


Figure 4.3.2 Tree with directional light propagation

In order to determine the input amounts of light to be sent in each direction for directional propagation, it is possible to integrate the sun's position in the sky over the period of one year (one tick = one year). Since the sun both rises and falls over the period of a day, and also changes positions in the sky over a year, we would need to solve a complex 2-dimensional integration [N05]. Websites exist for solar photovoltaic estimation, e.g. [SEH13], that give the amount of light that a square meter receives in a year (in MJ) as well as approximations for the amount of light that should be directed in each direction.

4.4 Code Outline

An overview of the classes that we implemented in our program are as follows:

- **Simulation** is the main class that establishes dimensions for the simulation and creates trees to grow. It is the control center of the model.
- **SimulationParams** stores environmental constants for the simulation such as direction of light, wind speed, and gravity.
- **Tree** is the class object for a single tree, i.e. a collection of segments.
- **TreeParams** stores constants for a given tree species such as the wood density, the branch angle factor, the area of leaves, and the photosynthesis growth exponent.
- **Segment** is the class object for a localized clump of cells in a tree. It is the basic building block of trees.
- **VoxelGrid** is a 3 dimensional grid that stores the amount of light available at any location in the simulation and prevents two segments from taking up the same space.
- **TreeCanvas** opens a window and draws the current tree in it using OpenGL.
- **Vector3** is a 3D vector class that extends Vector3f and adds a few functions that are not available in the Vector3f class.
- **CompareSegmentAge** is a class that implements Comparator to sort segments by their ages.

The full Java code listing can be found online, and Appendix E links to it.

5. Model Validation

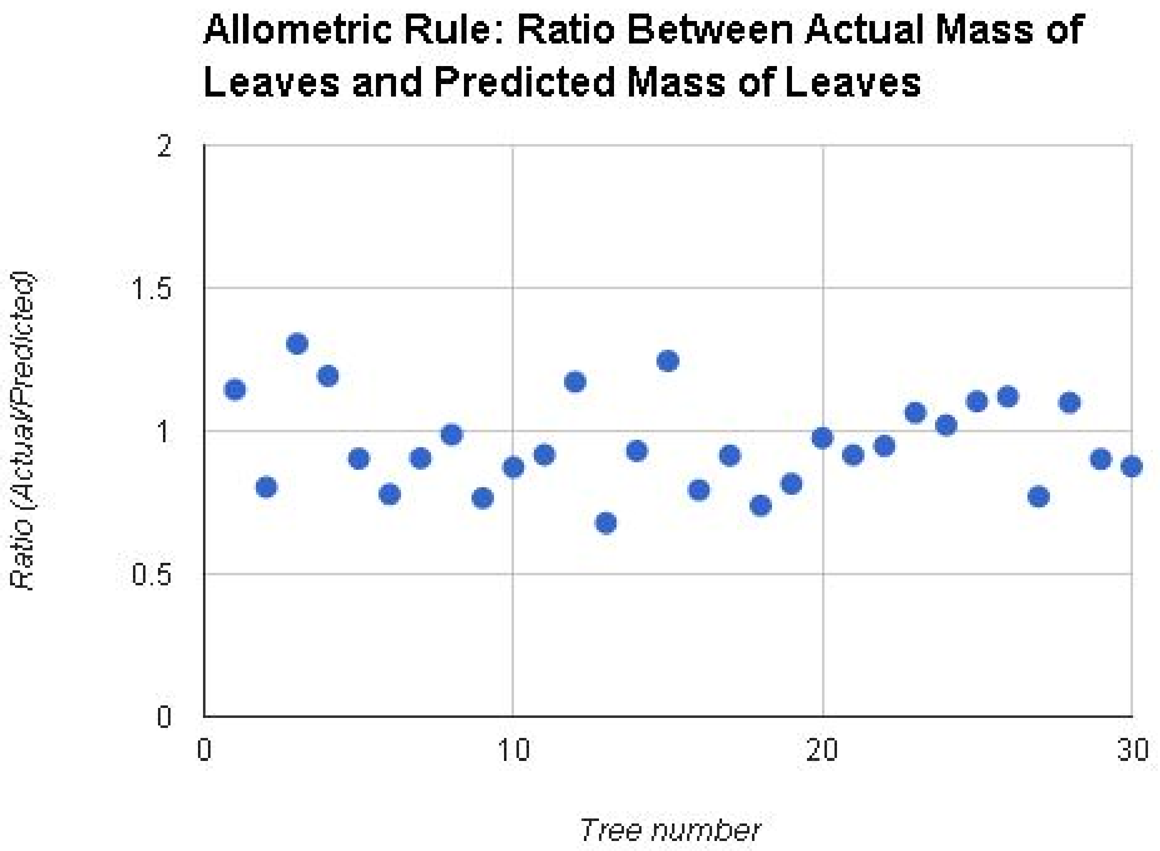
5.1 Allometric Rules

One of the main validations for our model is the confirmation that it exhibits allometric rules that hold true for all trees. Allometric rules are relationships between different quantitative values on a tree, but it is not known why all trees follow them. By showing that our model produces allometric rules that are observed in nature, we validate that our mathematical model is one possible explanation for these properties of growth. Although there are many allometric rules, we chose to test three different rules that stated concrete relationships between the mass of the leaves, mass of the stem, height, diameter, and total mass of the tree.

The first allometric rule that we tested is:

$$M_L = \beta * M_S^{\alpha}$$

where M_L is the mass of the leaves, M_S is the mass of the stem, and α and β are constants. According to a paper that we found, approximate values for α and β are 0.73 and 0.25, respectively [HN09]. In order to test that our trees exhibit this allometric rule, we calculated the mass of the leaves and the mass of the stem of the tree at age 60 for 30 different trees. Then, using the actual M_S of the tree, we calculated the predicted mass of the leaves. Data for the 30 trees can be found in Appendix D. If our tree in fact exhibits this allometric rule, then the ratio between the actual and predicted mass of the leaves would be approximately 1. The average ratio between the actual and predicted mass of the leaves in 30 trees had a mean of 0.955 with a standard deviation of 0.1615. Results are shown in Graph 5.1.1 below:



Graph 5.1.1 Confirmation for Allometric Rule 1

The graph shows that the model does in fact exhibit this allometric rule relatively closely.

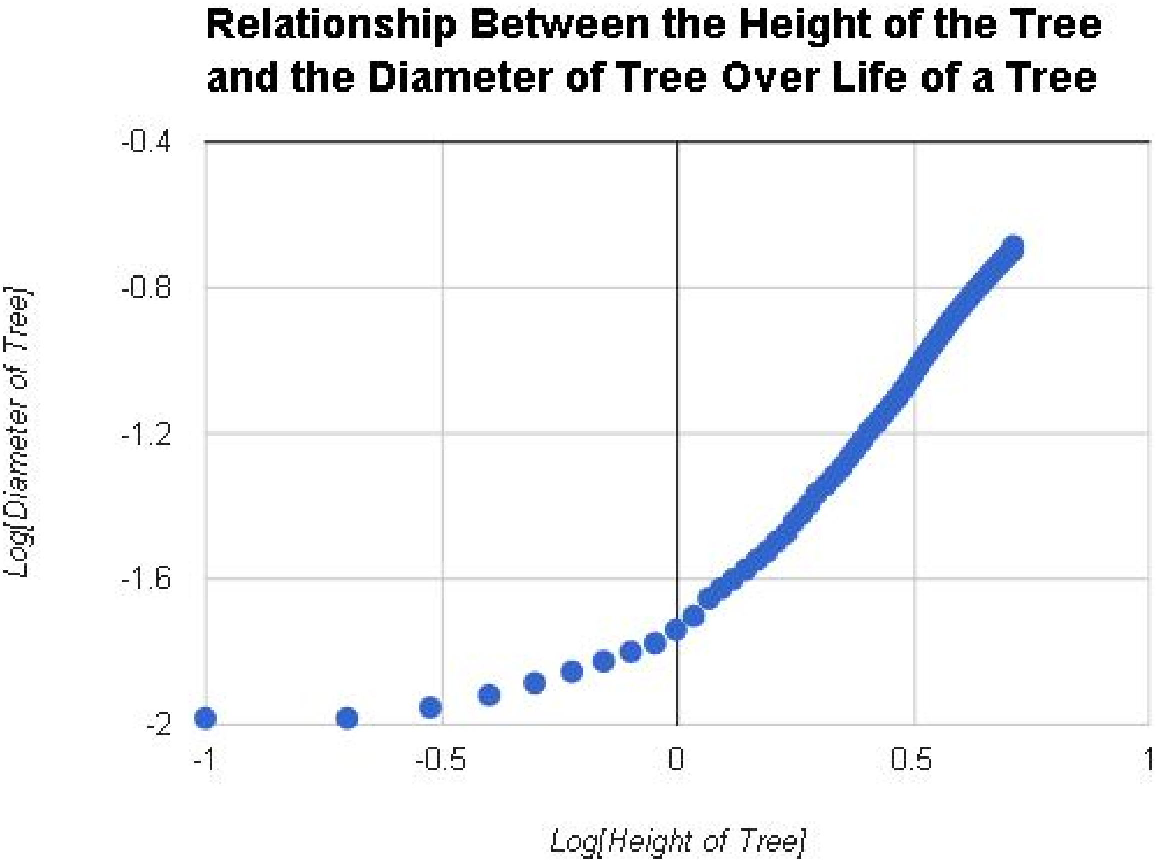
The next allometric rule that we chose to test is:

$$H \propto D^{2/3}$$

Where H is the height of the tree and D is the diameter of the tree. To test this allometric rule, we recorded the height of the tree and the diameter of the trunk of the tree for every year over a time period of 60 years. Data for this test can be found in Appendix D. Using logarithmic properties, the allometric rule can be changed to (for some constant c):

$$\log D = c + (3/2) \log H$$

The log of the height and diameter were calculated and plotted in a scatter plot. If our model exhibits this allometric rule, the scatter plot would be linear. After the first 10 years, our model exhibits a linear relationship with slope 3/2. Results are shown in Graph 5.1.2.



Graph 5.1.2 Confirmation for Allometric Rule 2

The final allometric rule we looked at was:

$$\text{Young trees: } M_L \propto M_T$$

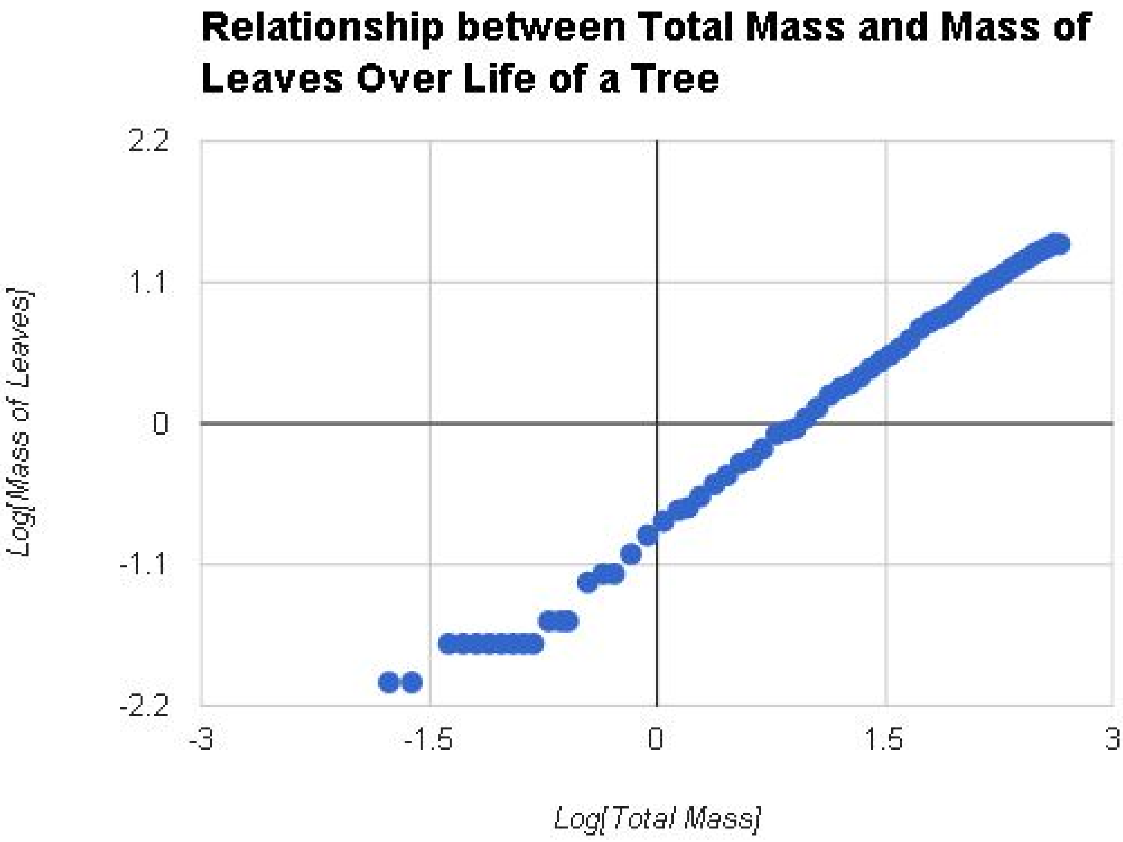
$$\text{Old Trees: } M_L \propto M_T^{0.75}$$

Where M_L is the mass of the leaves and M_T is the total mass of the tree. To test this allometric rule, we recorded the mass of the leaves and the total mass of the tree for every year over a time period of 60 years. Data for this test can be found in Appendix D. Using logarithmic properties, the allometric rules can be changed to:

Young trees: $\log M_L = c + \log M_T$

Old Trees: $\log M_L = c + 0.75 \log M_T$

The log of the height and diameter were calculated and plotted in a scatter plot. We set the cutoff date between young trees and old trees at age 30. If our model exhibits this allometric rule, the scatter plot would be linear. After the first 10 years, our model exhibits a linear relationship with slope 3/4. Results are shown in Graph 5.1.3.



Graph 5.1.3 Confirmation for Allometric Rule 3

5.2 Phototropism

Phototropism is an important feature of tree growth, and is exhibited by all trees, and plants in general. Positive phototropism is the propensity for plants to grow towards a source of light, providing that the light source is primarily from one direction. Some plants have light sensitive tissues that grow towards the light, and other simply exhibit phototropism due to the fact that tissues on the light side will grow more while those on the dark side will not.

Our model does not account for specialized tissues in plants that are designed to grow towards the light, but exhibits phototropism anyways. Using directional light propagation, it is easy to change input parameters for light so that the majority of light is coming from one direction. If the tree grows towards the light, then its center of mass will be skewed in the direction of the light. To test whether the tree does in fact grow in the direction of the light, we conducted hypothesis tests with trees growing in normal light and trees growing in light that is mainly coming from one direction.

Before we conducted a hypothesis test to confirm phototropism with abnormal light conditions in our model, we made sure to confirm that trees in normal light conditions grow upwards, or towards the light. Since the tree starts growing at position (5,5,0) in x,y,z coordinates, the tree in normal conditions should have a center of mass close to (5,5) in x,y coordinates. A Hypothesis test was conducted at a significance level of 0.001 testing the following null and alternative hypotheses:

$$H_o: u = 5.0$$

$$H_a: u \neq 5.0$$

where *u* is the mean x-coordinate of the final center of mass of the trees. Data was collected on a random sample of 30 simulated trees. Since the sample size was at least 30 and the conditions for conducting a hypothesis test were met, test statistics and P-values were calculated on the data. Collected data for the hypothesis test can be found in Appendix A. The resulting P-value was 0.37295. Since the P-value is greater than the significance level, we fail to reject the null, and do not have enough significant evidence to conclude that the tree consistently leans to a side.

The same hypothesis test as above was conducted on the mean y coordinate of the random sample of

30 simulated trees. Data for this test can also be found in Appendix A. The resulting P-value for the y-coordinate test was 0.31028. As in the previous test, we fail to reject the null and cannot conclude that the tree consistently leans to a side.

Since no evidence was found to conclude that the tree leans to one side in normal light conditions, we conducted an additional hypothesis test to prove phototropism in our model. We changed input light so that the majority of light was coming from the north. Since light was coming from the north, our model would only exhibit phototropism if the y-coordinate of the center of mass proved to be greater than 5.0 consistently. We conducted hypothesis tests at a significance level of 0.001 with the following null and alternative hypotheses:

$$H_o\colon u = 5.0$$

$$H_a\colon u > 5.0$$

where *u* is the mean y-coordinate of the final center of mass of the trees. Data was collected on a random sample of 30 simulated trees. With the conditions for hypothesis testing being met, we calculated test statistics and P-values using the data, which can be found in Appendix D. The resulting P-value was 2.017×10^{-6}. Since the P-value is less than the significance level, we can reject the null hypothesis and conclude with 99.9% confidence that the trees consistently grew into the light.

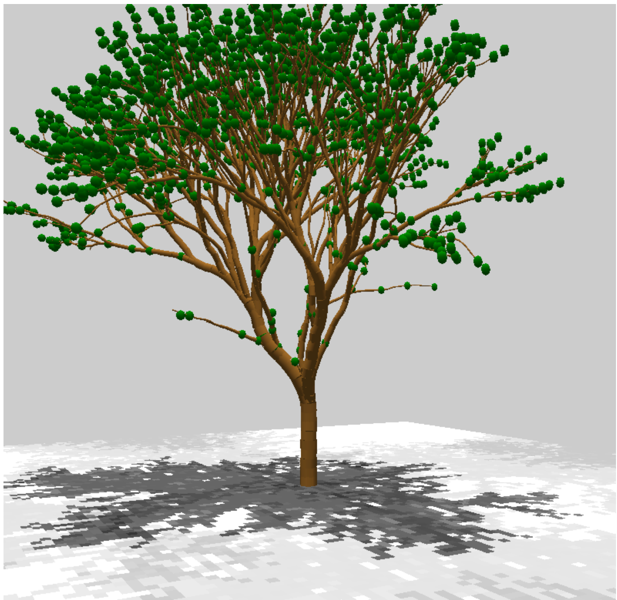


Figure 5.2.1 Tree in normal light conditions

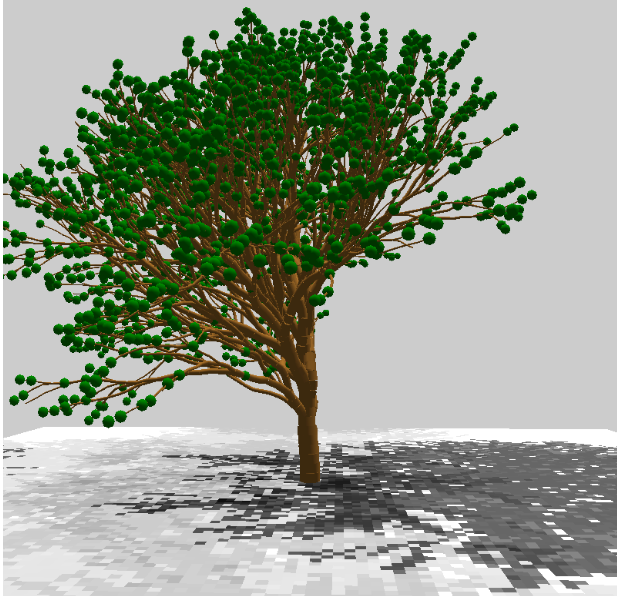


Figure 5.2.2 Tree with majority of light from north

5.3 Anemotropism

Trees in different environments experience different speeds of wind. In pleasant climates such as the tropics, trees can focus on growing in height rather than width; in harsher climates such as tundra, trees grow shorter in height but larger in width to stabilize themselves, preventing problems such as windthrow.

To test if our model responds to different wind conditions in the same manner, we conducted a hypothesis test. After checking the conditions for conducting a hypothesis test, we performed a 2-sample t test, comparing the z-coordinates of the center of mass of trees experiencing different wind speeds.

$$H_o\colon u_1 = u_2$$

$$H_a\colon u_1 > u_2$$

u_1 represents the z-coordinate of trees experiencing 20 m/s winds and u_2 represents the z-coordinate of trees experiencing 30 m/s winds. Data was collected on a random sample of 30 simulated trees. With the conditions for hypothesis testing being met, we calculated test statistics and P-values using the data, which can be found in Appendix B. The resulting P-value was 1.331×10^{-10}. Since the P-value is less than the significance level, we can reject the null hypothesis and conclude with 99.9% confidence that the trees consistently grew shorter when the wind speed felt was higher.

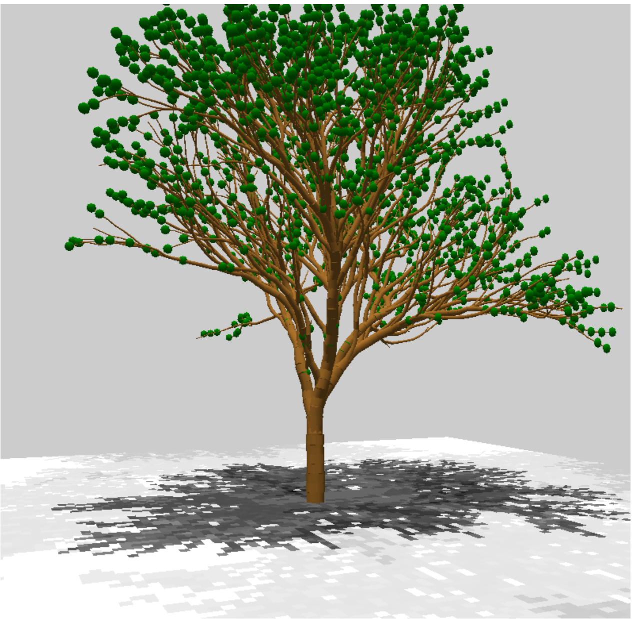


Figure 5.3.1 Tree Experiencing 20 m/s winds

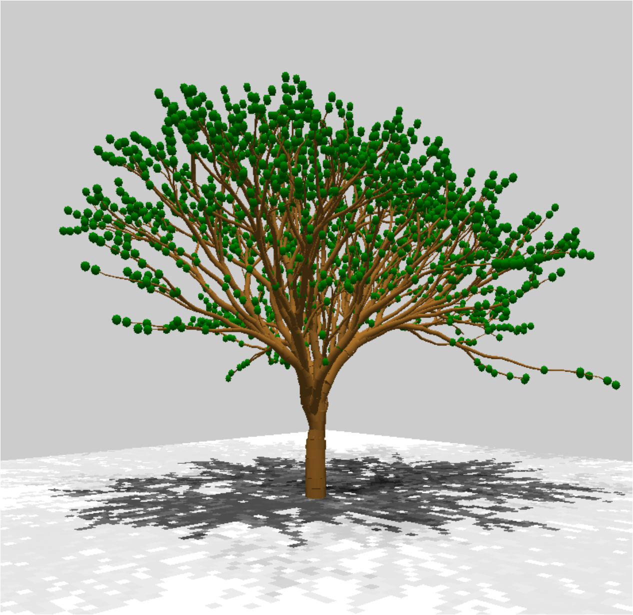


Figure 5.3.2 Tree Experiencing 30 m/s winds

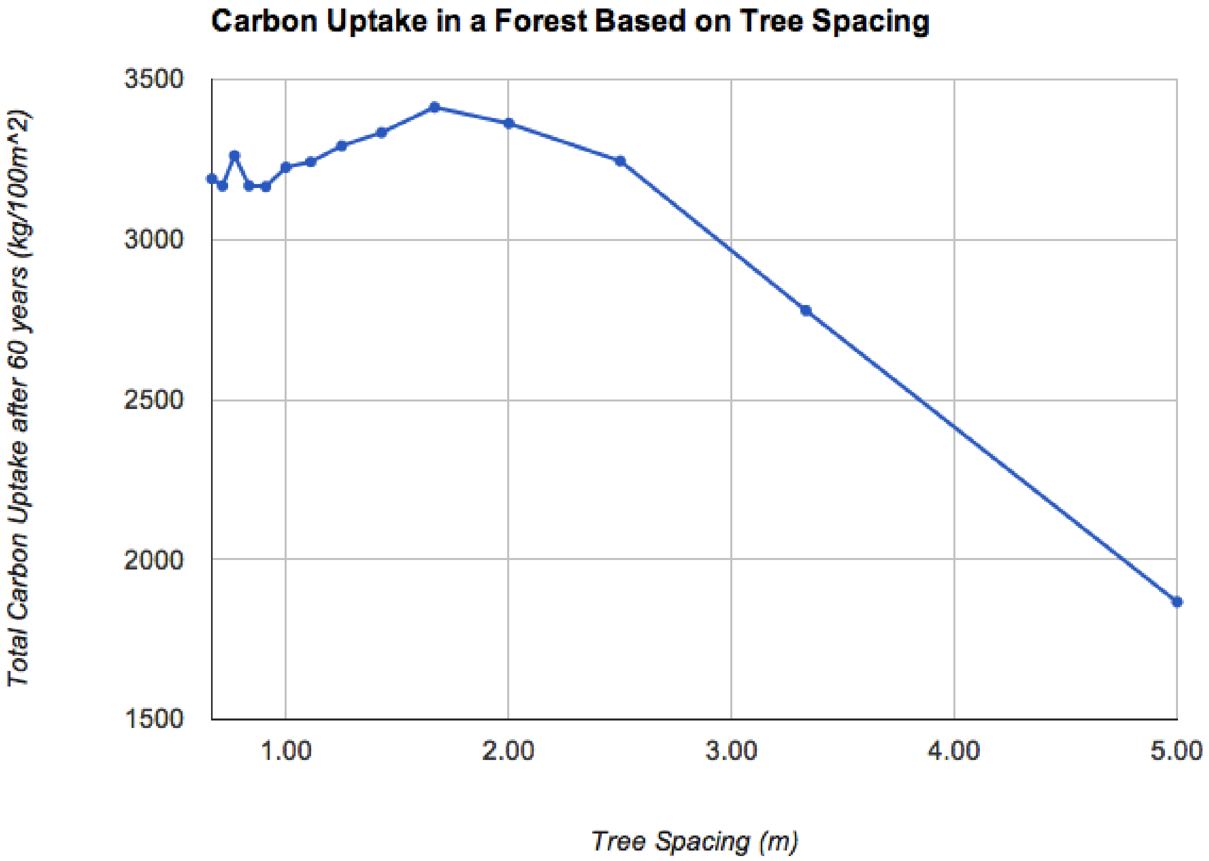
6. Applications

There are a number of possible applications for an efficient and accurate biological model of tree growth. We have only had time to explore one of them: optimizing tree planting spacing to maximize carbon sequestration.

6.1 Growing Trees for Maximum Carbon Sequestration

A common way of reducing carbon footprints is the sequestration of carbon through tree plantations. Studies have shown that tree plantations sequester carbon at up to 30 times the rate of old growth forests [D13]. Experiments to optimize tree planting conditions are extremely time-consuming, taking upwards of 15 years to complete. In many commercial situations, differences of a few percent in efficiency can make the difference between commercial success and failure. Being able to simulate and optimize tree growth and carbon uptake on tree plantations could make a significant difference in the commercial viability of carbon sequestration.

To find the optimal spacing between trees, we tested grids of 2x2 trees up to 15x15 trees in a 10x10 meter plot, finding the combined mass of all the trees after 60 years of growth. There are no edge effects in the simulation due to the implementation of a wrap-around world, which allows our simulation to essentially represent a small plot in an infinite forest. Data was collected on a random sample of 10 simulations in each setup, which can be found in Appendix C. We found that at a spacing of 1.67 m between trees, the combined mass was maximized.



Although the difference between spacings may seem insignificant, it can actually be a significant factor in the viability of tree plantations. For example, trees spaced 1.67 m apart sequester 51 kg more carbon per 100 m^2 than trees spaced 2 m apart. When extended to a square mile of trees, the difference between the two spacings results in an extra 1.3 kg of carbon sequestration per square mile. Although this is not accurate for any specific tree species, it demonstrates the potential of using a computer simulation to optimize planting conditions for carbon sequestration in a plantation.

7. Conclusion and Future Work

Based on the results of our model validation, we can conclude that we have successfully created a biologically accurate model of tree growth. The model combines the biological accuracy of a rule-based model with the graphics output of a image-based model. Additionally, the model exhibits critical emergent properties of tree growth without them being explicitly programmed into the code. These results provide one possible explanation for the reason that global allometric rules arise in tree growth without endocrine communication across the tree. In addition, the model can be used in a variety of different ways for practical applications. We demonstrate one potential application of the model by determining the optimal spacing for trees in a plantation to maximize carbon sequestration.

Despite our success, there is significant future work that could be done to improve the model and increase its accuracy. For example, our current program is a model for a general deciduous tree. We have a set of parameters that define characteristics such as the a leaf's mass and area; changing those values would ideally give us a specific deciduous tree, or even a coniferous tree. Being able to create different types of trees would make forest modeling more realistic, since forests have more than one type of tree.

Another addition that could be made to the model is the consideration of water. While our model currently focuses on light availability as the limiting factor, it could be improved by considering water availability in the growth process. This would make our model more accurate for trees in environments such as the desert, where water is the limiting factor for growth. Adding the consideration of water to the model would also allow us to model important factors of growth such as evaporation and transpiration. The addition of water may also require the modeling of root systems.

8. Acknowledgements

We'd like to thank Jason DeWitte, our sponsor teacher, who taught us a great deal about biology in general, as well as general tree information and some defining factors of tree growth.

Additionally, we would like to thank Dr. Sean Hammond from the University of New Mexico. Sean Hammond does work in tree modeling and was very helpful in defining the importance of allometric rules for us. He also gave us advice on general characteristics of tree growth and the ways that trees apportion their carbon. Dr. Hammond is currently helping us submit our model for presentation at a botany conference and eventually publication.

Lastly, we would like to thank Professor David Hanson from the University of New Mexico. Professor Hanson helped to offer ideas for the project in general, and gave advice on the focus of light propagation through the canopy.

9. References

[ASG+11] Agusti, J., E. M. Sehr, T. Greb, T. Sieberer, C. A. Beveridge, P. B. Brewer, E. A. Dun, K. Ljung, P. Sanchez, M. Schwarz, and S. Herold. "Strigolactone signaling is required for auxin-dependent stimulation of secondary growth in plants." *Proceedings of the National Academy of Sciences* 108.50 (2011): 20242-20247.

[A07] Albers, Katharina. “Allometric Scaling Laws in Nature pt. 3.” *Gute Ideen in der theoretischen Systembiologie*: 77, July 2007.

[BW06] Barrera, E. De La, and Walter, H. S., “Wind effects on leaf morphology for the mangrove *Conocarpus erecta* at an oceanic island from the Mexican Pacific Ocean.” *Revista Chilena de Historia Natural* 79(4): 451-463, 2006.

[CN98] Campbell, G. S. and Norman, J. M. *An Introduction to Environmental Biophysics, Second Edition*. Springer-Verlag, New York, 1998.

[CBGB91] Collatz, C. J., Ball, J. T., Grivet, C., and Berry, J. A., “Physiological and environmental regulation of stomatal conductance, photosynthesis, and transpiration: a model that includes a laminar boundary layer.” *Agric. For. Meteorol.* 54:107–136, 1991.

[D13] Dombro, D., “How much carbon does a tropical tree sequester?” United Nations Environment Programme: Plant for the planet, Billion tree campaign. http://www.tree-nation.com/public/documents/tropical_tree_sequestration_co2_tree-nation.pdf, accessed 12 Dec 2013.

[EKVS11] Evers, J. B., van der Krol, A. R., Vos, J., and Struik, P. C., “Understanding shoot branching by modelling form and function.” *Trends in Plant Science* 2011: 1–4.

[FBEE] *Forest Biology and Ecology for Educators*, ch. 4, “Water Relations.” Virginia Tech Department of Forest Resources and Environmental Education. http://dendro.cnre.vt.edu/forbioeco/htmltext/chapter4.htm, accessed 5 Jan 2014.

[FKKC10] Finlayson, S. A., Krishnareddy, S. R., Kebrom, T. H., and Casal, J. J., “Phytochrome regulation of branching in *Arabidopsis*.” *Plant Physiology* 152(4): 1914–1927, 2010.

[HN09] Hammond, S. T. and Niklas, K. J., “Emergent properties of plants competing in silico for space and light: seeing the tree from the forest.” *American J. of Botany* 96(8): 1430–1444, 2009.

[MH25] Green, R. E. and McCauley, C. J., editors. *Machinery's Handbook, 25th Edition*. Industrial Press, Inc., New York, 1996.

[Mon77] Monteith, J. L., "Climate and the efficiency of crop production in England." *Phil. Trans. R. Soc. Lond. B.* 281: 277–294, 1977.

[NZSS13] Neissner, M., Zollhofer, M, Shahram, I., and Stamminger, M., "Real-time 3D Reconstruction at Scale using Voxel Hashing." *ACM Trans. on Graphics* 32(6), Nov. 2013.

[NFD07] Neubert, B., Franken, T., & Deussen, O. (2007). Approximate image-based tree-modeling using particle flows. *ACM Transactions on Graphics*, *26*(3), 88.

[N99] Niklas, K. (1999). Evolutionary walks through a land plant morphospace. *Journal of Experimental Botany*, *50*(330), 39-52.

[N05] Nobel, P. S., *Physicochemical and Environmental Plant Physiology, 3rd Edition*. Academic Press/Elsevier, Burlington, MA, 2005.

[PVN05] Peramaki, M., Vesala, T., & Nikinmaa, E. (2005). Modeling the dynamics of pressure propagation and diameter variation in tree sapwood. *Tree Physiology*, *25*(9), 1091-1099

[P09] Persson, M. *Minecraft*. Video game published by Mojang, Stockholm, Sweden, 2009.

[PAR14] "Photosynthetically Active Radiation." *Wikipedia*, http://en.wikipedia.org/wiki/Photosynthetically_active_radiation, accessed 2 Jan 2014.

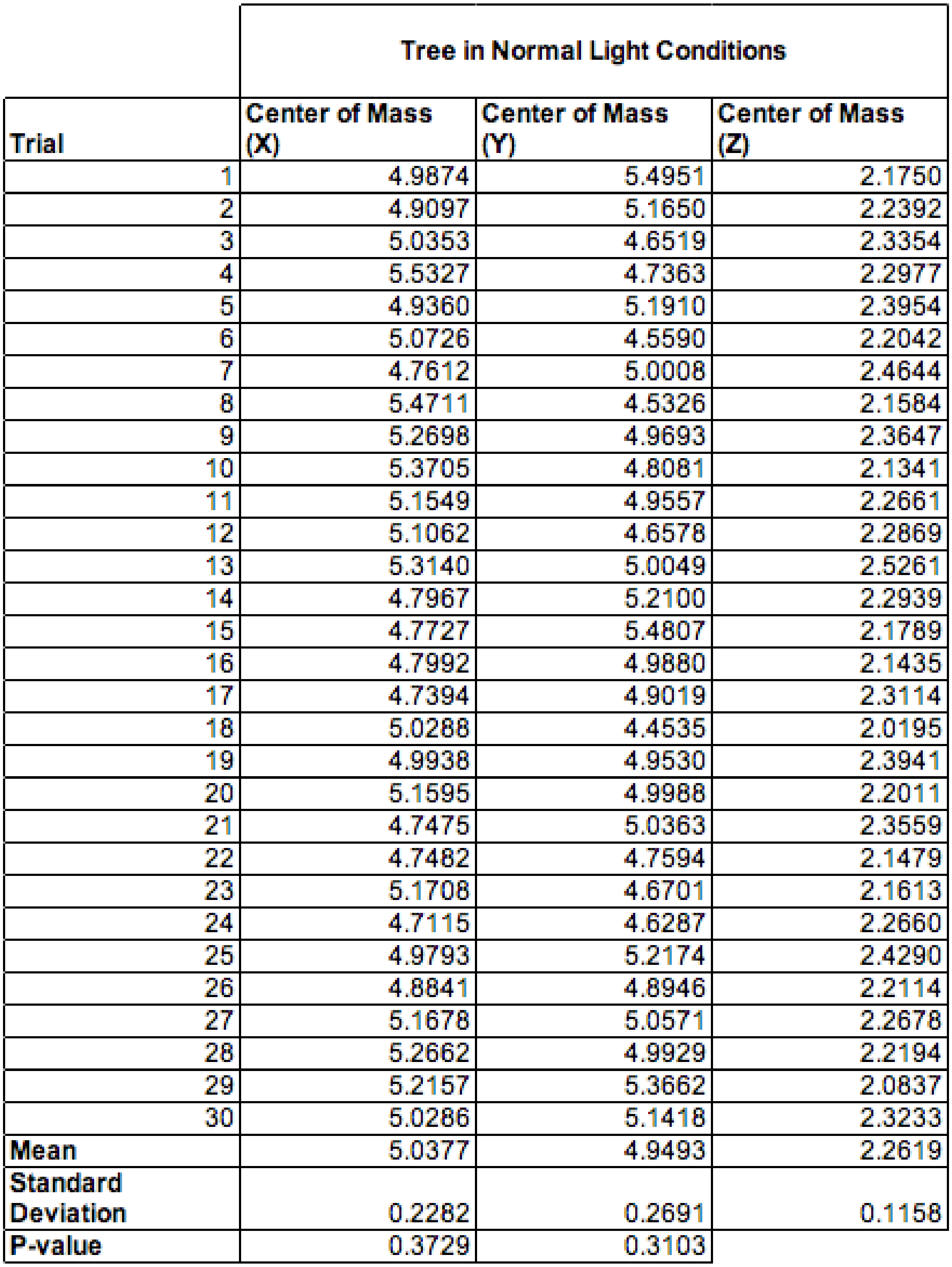
[SEH13] *Solar Electricity Handbook* online irradiance calculator. http://www.solarelectricityhandbook.com/solar-calculator.html, accessed 27 Feb 2014.

[TZW+07] Tan, P., Zeng, G., Wang, J., Kang, S. B., & Quan, L., "Image-based tree modeling." *ACM Transactions on Graphics*,*26*(3), 87, 2007.

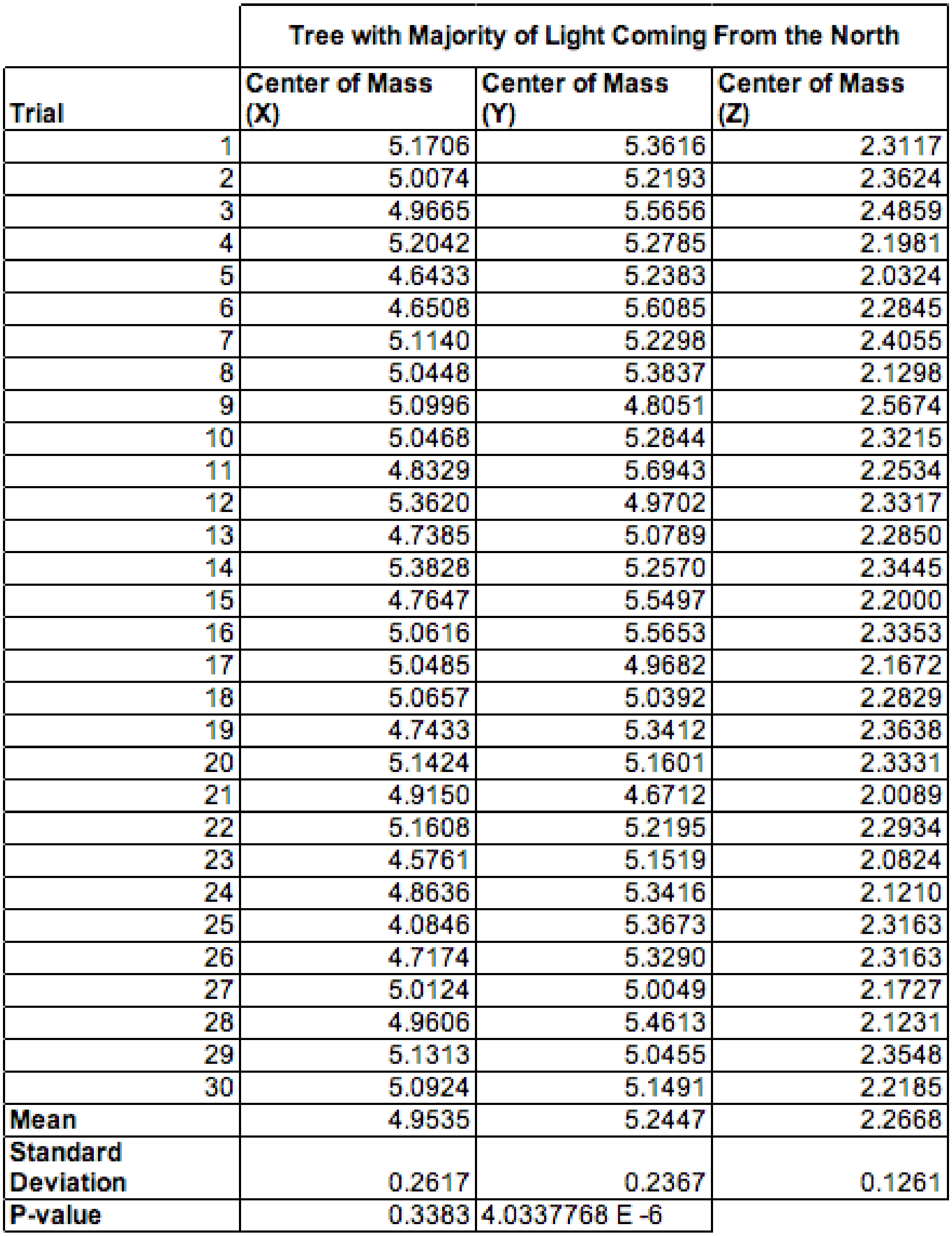
[TJT+01] Tolbert, V., Joslin, J., Thornton, F., Bock, B., Pettry, D., Bandaranayake, W., Tyler, D., Houston, A., and Schoenholtz, S., "Biomass Crop Production: Benefits for Soil Quality and Carbon Sequestration." 2001. http://csmb.ornl.gov/~webworks/cppr/y2001/pres/113727.pdf, accessed 14 Mar 2014.

[V09] Vogel, S., "Leaves in the lowest and highest winds: temperature, force and shape." *New Phytologist* 183(1): 13–26, July 2009.

Appendix A: Data for Phototropism Analysis

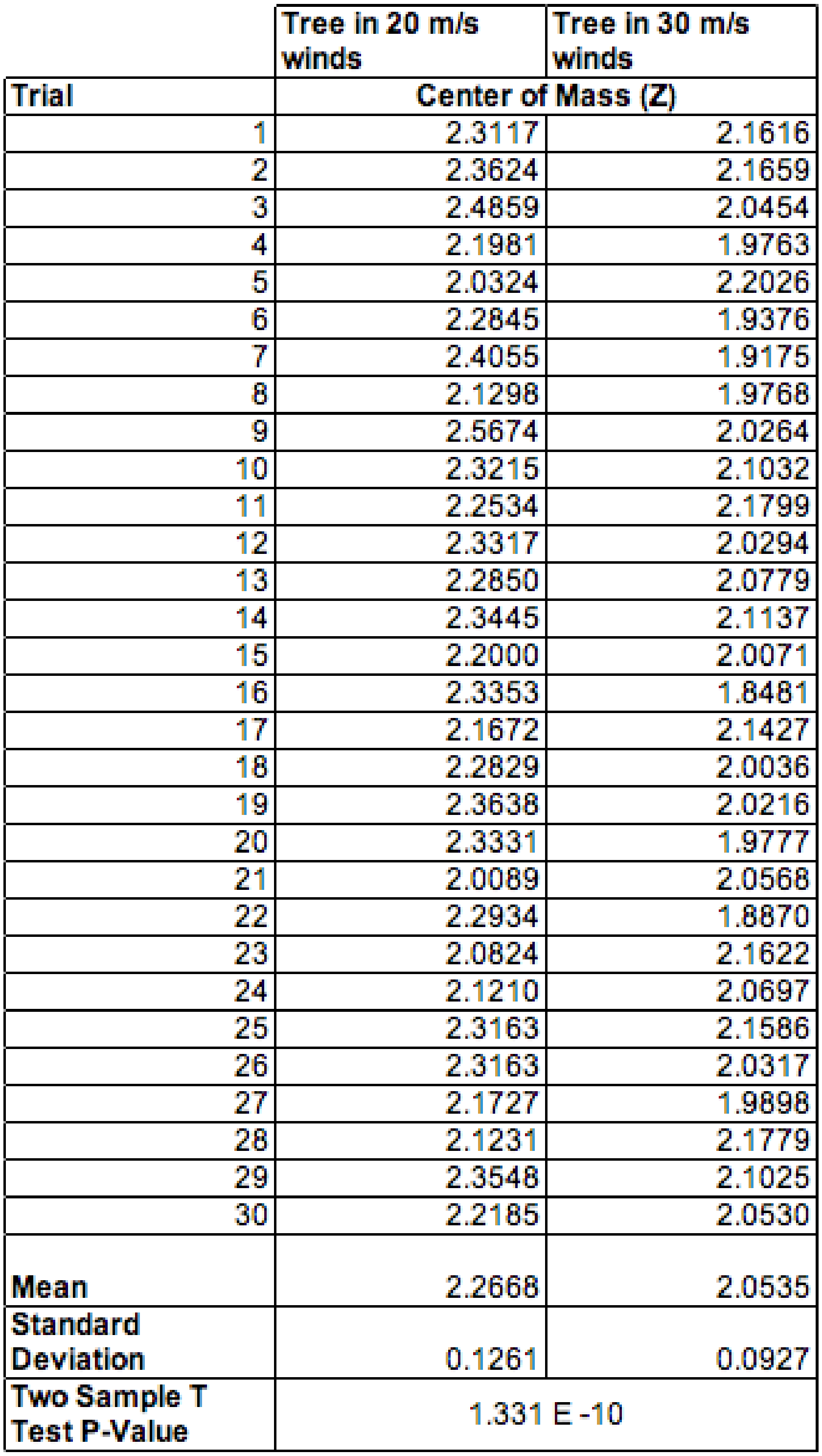


	Tree in Normal Light Conditions		
Trial	Center of Mass (X)	Center of Mass (Y)	Center of Mass (Z)
1	4.9874	5.4951	2.1750
2	4.9097	5.1650	2.2392
3	5.0353	4.6519	2.3354
4	5.5327	4.7363	2.2977
5	4.9360	5.1910	2.3954
6	5.0726	4.5590	2.2042
7	4.7612	5.0008	2.4644
8	5.4711	4.5326	2.1584
9	5.2698	4.9693	2.3647
10	5.3705	4.8081	2.1341
11	5.1549	4.9557	2.2661
12	5.1062	4.6578	2.2869
13	5.3140	5.0049	2.5261
14	4.7967	5.2100	2.2939
15	4.7727	5.4807	2.1789
16	4.7992	4.9880	2.1435
17	4.7394	4.9019	2.3114
18	5.0288	4.4535	2.0195
19	4.9938	4.9530	2.3941
20	5.1595	4.9988	2.2011
21	4.7475	5.0363	2.3559
22	4.7482	4.7594	2.1479
23	5.1708	4.6701	2.1613
24	4.7115	4.6287	2.2660
25	4.9793	5.2174	2.4290
26	4.8841	4.8946	2.2114
27	5.1678	5.0571	2.2678
28	5.2662	4.9929	2.2194
29	5.2157	5.3662	2.0837
30	5.0286	5.1418	2.3233
Mean	5.0377	4.9493	2.2619
Standard Deviation	0.2282	0.2691	0.1158
P-value	0.3729	0.3103	



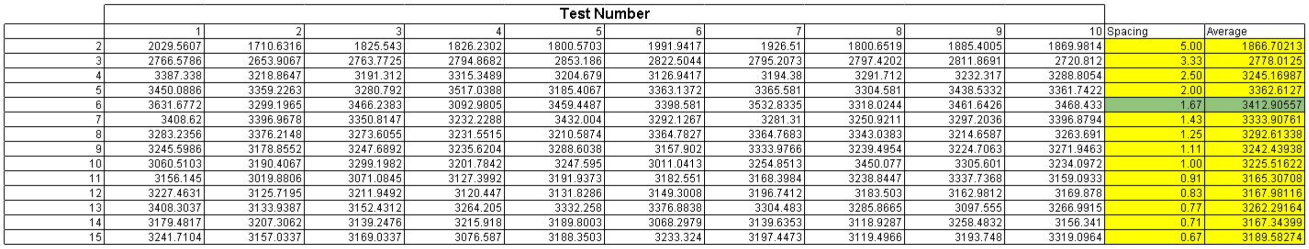
	Tree with Majority of Light Coming From the North		
Trial	Center of Mass (X)	Center of Mass (Y)	Center of Mass (Z)
1	5.1706	5.3616	2.3117
2	5.0074	5.2193	2.3624
3	4.9665	5.5656	2.4859
4	5.2042	5.2785	2.1981
5	4.6433	5.2383	2.0324
6	4.6508	5.6085	2.2845
7	5.1140	5.2298	2.4055
8	5.0448	5.3837	2.1298
9	5.0996	4.8051	2.5674
10	5.0468	5.2844	2.3215
11	4.8329	5.6943	2.2534
12	5.3620	4.9702	2.3317
13	4.7385	5.0789	2.2850
14	5.3828	5.2570	2.3445
15	4.7647	5.5497	2.2000
16	5.0616	5.5653	2.3353
17	5.0485	4.9682	2.1672
18	5.0657	5.0392	2.2829
19	4.7433	5.3412	2.3638
20	5.1424	5.1601	2.3331
21	4.9150	4.6712	2.0089
22	5.1608	5.2195	2.2934
23	4.5761	5.1519	2.0824
24	4.8636	5.3416	2.1210
25	4.0846	5.3673	2.3163
26	4.7174	5.3290	2.3163
27	5.0124	5.0049	2.1727
28	4.9606	5.4613	2.1231
29	5.1313	5.0455	2.3548
30	5.0924	5.1491	2.2185
Mean	4.9535	5.2447	2.2668
Standard Deviation	0.2617	0.2367	0.1261
P-value	0.3383	4.0337768 E -6	

Appendix B: Data for Anemotropism Analysis



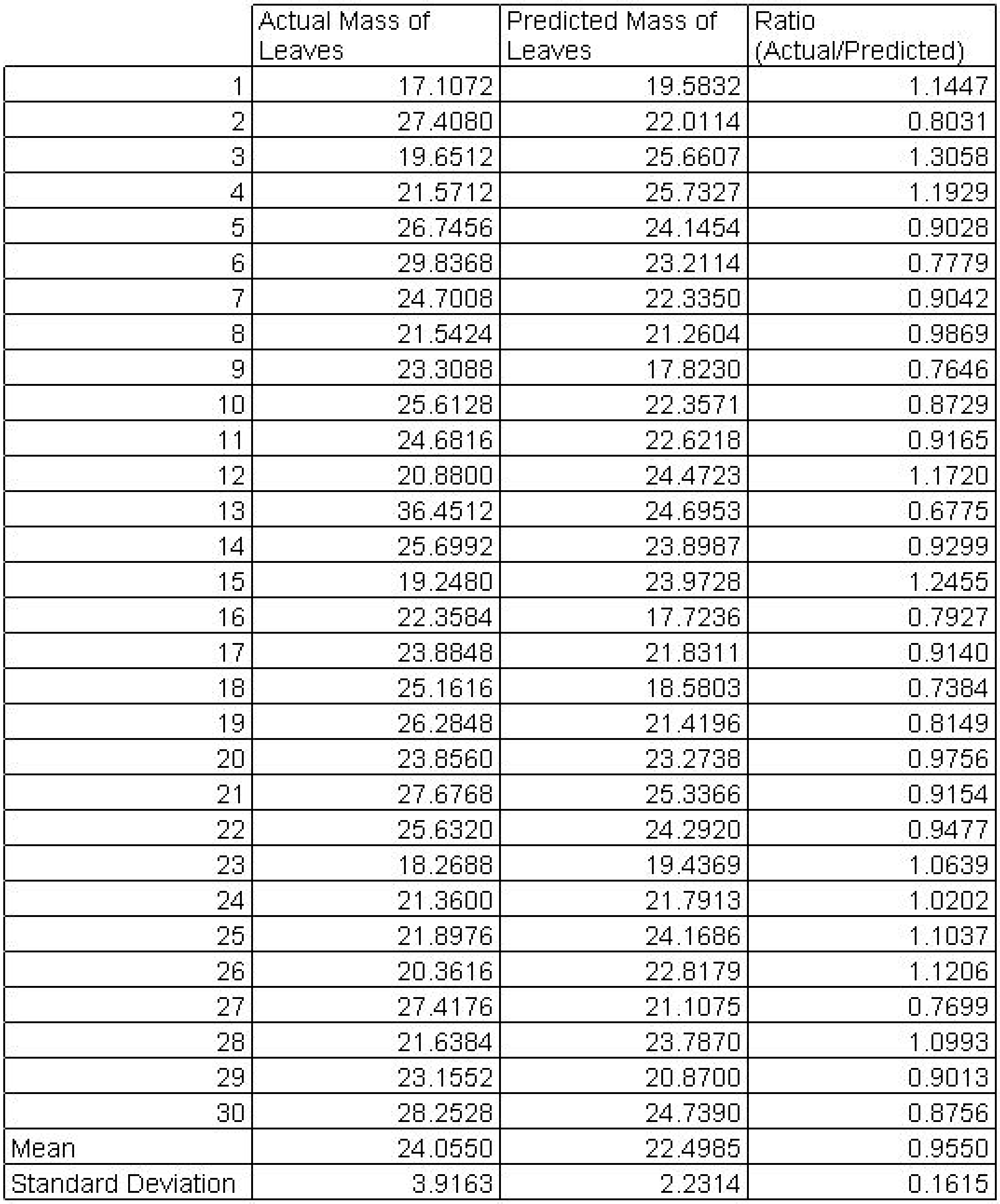
	Tree in 20 m/s winds	Tree in 30 m/s winds
Trial	Center of Mass (Z)	
1	2.3117	2.1616
2	2.3624	2.1659
3	2.4859	2.0454
4	2.1981	1.9763
5	2.0324	2.2026
6	2.2845	1.9376
7	2.4055	1.9175
8	2.1298	1.9768
9	2.5674	2.0264
10	2.3215	2.1032
11	2.2534	2.1799
12	2.3317	2.0294
13	2.2850	2.0779
14	2.3445	2.1137
15	2.2000	2.0071
16	2.3353	1.8481
17	2.1672	2.1427
18	2.2829	2.0036
19	2.3638	2.0216
20	2.3331	1.9777
21	2.0089	2.0568
22	2.2934	1.8870
23	2.0824	2.1622
24	2.1210	2.0697
25	2.3163	2.1586
26	2.3163	2.0317
27	2.1727	1.9898
28	2.1231	2.1779
29	2.3548	2.1025
30	2.2185	2.0530
Mean	2.2668	2.0535
Standard Deviation	0.1261	0.0927
Two Sample T Test P-Value	1.331 E -10	

Appendix C: Data for Maximization of Carbon Sequestration in a Plantation Analysis



	Test Number											
	1	2	3	4	5	6	7	8	9	10	Spacing	Average
2	2029.5607	1710.6316	1825.543	1826.2302	1800.5703	1991.9417	1926.51	1800.6519	1885.4005	1869.9814	5.00	1866.70213
3	2766.5786	2653.9067	2763.7725	2794.8682	2853.186	2822.5044	2795.2073	2797.4202	2811.8691	2720.812	3.33	2778.0125
4	3387.338	3218.8647	3191.312	3315.3489	3204.679	3126.9417	3194.38	3291.712	3232.317	3288.8054	2.50	3245.16987
5	3450.0886	3359.2263	3280.792	3517.0388	3185.4067	3363.1372	3365.581	3304.581	3438.5332	3361.7422	2.00	3362.6127
6	3631.6772	3299.1965	3466.2383	3092.9805	3459.4487	3398.581	3532.8335	3318.0244	3461.6426	3468.433	1.67	3412.90557
7	3408.62	3396.9678	3350.8147	3232.2288	3432.004	3292.1267	3281.31	3250.9211	3297.2036	3396.8794	1.43	3333.90761
8	3283.2356	3376.2148	3273.6055	3231.5515	3210.5874	3364.7827	3364.7683	3343.0383	3214.6587	3263.691	1.25	3292.61338
9	3245.5986	3178.8552	3247.6892	3235.6204	3288.6038	3157.902	3333.9766	3239.4954	3224.7063	3271.9463	1.11	3242.43938
10	3060.5103	3190.4067	3299.1982	3201.7842	3247.595	3011.0413	3254.8513	3450.077	3305.601	3234.0972	1.00	3225.51622
11	3156.145	3019.8806	3071.0845	3127.3992	3191.9373	3182.551	3168.3984	3238.8447	3337.7368	3159.0933	0.91	3165.30708
12	3227.4631	3125.7195	3211.9492	3120.447	3131.8286	3149.3008	3196.7412	3183.503	3162.9812	3169.878	0.83	3167.98116
13	3408.3037	3133.9387	3152.4312	3264.205	3332.258	3376.8838	3304.483	3285.8665	3097.555	3266.9915	0.77	3262.29164
14	3179.4817	3207.3062	3139.2476	3215.918	3189.8003	3068.2979	3139.6353	3118.9287	3258.4832	3156.341	0.71	3167.34399
15	3241.7104	3157.0337	3169.0337	3076.587	3188.3503	3233.324	3197.4473	3119.4966	3193.748	3319.0964	0.67	3189.58274

Appendix D: Data for Allometric Rules Analysis



	Actual Mass of Leaves	Predicted Mass of Leaves	Ratio (Actual/Predicted)
1	17.1072	19.5832	1.1447
2	27.4080	22.0114	0.8031
3	19.6512	25.6607	1.3058
4	21.5712	25.7327	1.1929
5	26.7456	24.1454	0.9028
6	29.8368	23.2114	0.7779
7	24.7008	22.3350	0.9042
8	21.5424	21.2604	0.9869
9	23.3088	17.8230	0.7646
10	25.6128	22.3571	0.8729
11	24.6816	22.6218	0.9165
12	20.8800	24.4723	1.1720
13	36.4512	24.6953	0.6775
14	25.6992	23.8987	0.9299
15	19.2480	23.9728	1.2455
16	22.3584	17.7236	0.7927
17	23.8848	21.8311	0.9140
18	25.1616	18.5803	0.7384
19	26.2848	21.4196	0.8149
20	23.8560	23.2738	0.9756
21	27.6768	25.3366	0.9154
22	25.6320	24.2920	0.9477
23	18.2688	19.4369	1.0639
24	21.3600	21.7913	1.0202
25	21.8976	24.1686	1.1037
26	20.3616	22.8179	1.1206
27	27.4176	21.1075	0.7699
28	21.6384	23.7870	1.0993
29	23.1552	20.8700	0.9013
30	28.2528	24.7390	0.8756
Mean	24.0550	22.4985	0.9550
Standard Deviation	3.9163	2.2314	0.1615

Height	**Diameter**
0.10	0.01
0.20	0.01
0.30	0.01
0.40	0.01
0.50	0.01
0.60	0.01
0.70	0.01
0.80	0.02
0.90	0.02
0.99	0.02
1.08	0.02
1.17	0.02
1.24	0.02
1.31	0.03
1.40	0.03
1.47	0.03
1.55	0.03
1.62	0.03
1.70	0.03
1.77	0.04
1.84	0.04
1.90	0.04
1.97	0.04
2.06	0.05
2.15	0.05
2.23	0.05
2.31	0.05
2.39	0.06
2.47	0.06
2.55	0.06
2.65	0.07

2.74	0.07
2.84	0.08
2.93	0.08
3.01	0.08
3.09	0.09
3.16	0.09
3.24	0.10
3.32	0.10
3.40	0.11
3.48	0.11
3.56	0.12
3.65	0.12
3.73	0.12
3.80	0.13
3.90	0.13
3.99	0.14
4.08	0.14
4.17	0.15
4.27	0.16
4.37	0.16
4.47	0.17
4.56	0.17
4.65	0.18
4.75	0.18
4.85	0.19
4.95	0.19
5.04	0.20
5.14	0.20
5.14	0.21

Total Mass	Mass of Leaves
0.02	0.01
0.02	0.01
0.04	0.02
0.05	0.02
0.07	0.02
0.08	0.02
0.09	0.02
0.11	0.02
0.13	0.02
0.15	0.02
0.19	0.03
0.23	0.03
0.26	0.03
0.35	0.06
0.44	0.07
0.52	0.07
0.68	0.10
0.87	0.13
1.11	0.17
1.39	0.21
1.61	0.22
1.93	0.27
2.39	0.34
2.89	0.39
3.54	0.49
4.21	0.53
4.96	0.63
6.11	0.83
7.19	0.87
8.20	0.91
9.58	1.10
11.39	1.32
13.64	1.65

16.17	1.89
18.63	2.02
21.73	2.29
25.46	2.69
29.70	3.06
34.29	3.41
39.83	3.87
46.13	4.49
54.04	5.49
62.82	6.17
71.91	6.66
81.55	7.09
91.94	7.81
104.62	9.02
118.53	10.06
133.83	11.50
151.22	12.39
170.31	13.32
192.10	14.67
215.66	16.18
242.84	17.77
270.59	19.00
302.32	20.91
335.82	22.17
371.68	23.45
410.27	25.12
448.08	24.97

Appendix E: Code Listing

Our final single-computer Java code as it was at the time of filing this final report is at:
http://goo.gl/GUUR2b

Appendix F: TreeParams Code

```
public class TreeParams {
	// Size of random vector added to unit vector of parent segment's 
direction
	// to determine direction of child segment.
	public float branchAngleFactor;
	private float youngBranchAngleFactor;
	private float oldBranchAngleFactor;
	private float segmentLength; //meters
	private float woodDensity; //kg/m3
	private float dragCoefficient;
	private float massOfLeaf; //kg
	private float leafArea; //m^2
	private float fractionAbsorbed; //fraction of light absorbed by the 
canopy, fs
	private float efficiency; //efficiency of leaves, e
	private float rootApportion; //proportion of carbon that goes towards 
roots
	private int maxAge; //maximum age of tree in years
	private float woodWaterFraction;
	private float leafDryMass;
	private float stressProportion;
	private int maxLeafAge;

	//constructor: private so only created by functions below
	private TreeParams () {

	}

	//creates tree params for a general deciduous tree
	public static TreeParams createGeneralDeciduousTreeParams() {
		TreeParams t = new TreeParams();
		t.youngBranchAngleFactor = 0.01f;
		t.oldBranchAngleFactor = 0.05f;
```

```
    t.segmentLength = .1f;
    t.dragCoefficient = 0.5f;
    t.woodDensity = 900; //oak
    t.massOfLeaf = 0.0096f;
    t.leafArea = 0.0032f;
    t.fractionAbsorbed = 0.8f;
    t.efficiency = .0015f;
    t.rootApportion = 0.4f;
    t.maxAge = 60; //years
    t.woodWaterFraction = 0.59f; //fraction of wood that is water
    t.leafDryMass = 0.00192f; //fraction of leaf that is water: 0.8.
    t.stressProportion = 950000000;
    t.maxLeafAge = 1;
    return t;
}

public static TreeParams createAmericanSycamoreTreeParams() {
    TreeParams t = new TreeParams();
    t.youngBranchAngleFactor = 0.01f;
    t.oldBranchAngleFactor = 0.06f;
    t.segmentLength = .2f;
    t.dragCoefficient = 0.5f;
    t.woodDensity = 460;
    t.massOfLeaf = 0.048f;
    t.leafArea = 0.16f;  //16 leaves
    t.fractionAbsorbed = 0.8f;
    t.efficiency = .0015f;
    t.rootApportion = 0.4f;
    t.maxAge = 150; //years
    t.woodWaterFraction = 0.59f; //fraction of wood that is water
    t.leafDryMass = 0.0096f; //fraction of leaf that is water: 0.8.
    t.stressProportion = 950000000;
    t.maxLeafAge = 1;
    return t;
}
```

```
public static TreeParams createSugarMapleTreeParams() {
      TreeParams t = new TreeParams();
      t.youngBranchAngleFactor = 0.02f;
      t.oldBranchAngleFactor = 0.05f;
      t.segmentLength = .295f;
      t.dragCoefficient = 0.5f;
      t.woodDensity = 676;
      t.massOfLeaf = 0.048f;
      t.leafArea = 0.16f;  //16 leaves
      t.fractionAbsorbed = 0.8f;
      t.efficiency = .0015f;
      t.rootApportion = 0.4f;
      t.maxAge = 150; //years
      t.woodWaterFraction = 0.59f; //fraction of wood that is water
      t.leafDryMass = 0.0096f; //fraction of leaf that is water: 0.8.
      t.stressProportion = 2140000000;
      t.maxLeafAge = 1;
      return t;
}
...
}
```