GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Bachelor's Thesis

submitted in partial fulfillment of the
requirements for the course "Applied Computer Science"

Simulation of Ivy: Growth, 3D-Structure, Light interception and protective functions in facade greening

Isaac Bregulla

Institute of Computer Science

Bachelor's and Master's Theses
of the Center for Computational Sciences
at the Georg-August-Universität Göttingen

07. February 2025

Georg-August-Universität Göttingen
Institute of Computer Science

Goldschmidtstraße 7
37077 Göttingen
Germany

☎ +49 (551) 39-172000
FAX +49 (551) 39-14403
✉ office@informatik.uni-goettingen.de
www.informatik.uni-goettingen.de

First Supervisor: Prof. Dr. Winfried Kurth
Second Supervisor: Prof. Dr. Wolfgang Rohe

I hereby declare that I have written this thesis independently without any help from others and without the use of documents or aids other than those stated. I have mentioned all used sources and cited them correctly according to established academic citation rules.

Göttingen, 07. February 2025

Abstract

The plant Hedera helix is a perennial vine in most of Europe, which is well known for its function as facade greenery. Based on this common use, this paper seeks to simulate the plant behaviour on a flat surface, conceived as a wall of a building. The software GroIMP provided by the University of Göttingen facilitates this study, the inherent programming language XL will be the basis of this work and further expounded on in this paper. The complexity of plant growth and the amount of external forces on the growth behaviour make the undertaking of simulating a realistic plant very difficult, some of these factors this paper will seek to cover, prominently the light interception of leaves and their response. Subsequently, this thesis will cover relevant scientific papers on the topic of ivy morphology and plant patterns, including data gathered for this work to complete an extensive summary of Hedera helix. These guard rails will set the prerequisites for a realistic plant model.

Contents

Chapter 1

Introduction

Climate change poses major challenges for us today, the once jointly decided upon Paris Agreements are again under attack, all the while the threshold of 1.5°C set out in 2015 will soon be breached with estimates concluding that a temperature increase of 1.45°C compared to the pre-industrial era has already taken place. The results are conclusive with health, food security and weather in jeopardy [1], additionally the extinction of species, which is favored by climate change.

These threats are countered by international and national initiatives, which aim to reach climate goals and work to counter health risks to a rising urban population. Some objectives are met by governments encouraging the growth of common ivy as a means to cut energy consumption.

The advantages of using ivy as insulation are primarily felt during winter periods and extreme weather e.g. windy or rainy, in which the ivy inhibits a wet facade and can facilitate warm air pockets which in turn lead to less ablation thereof.

Stronger coverage seems to correlate with better insulation, during the winter months the difference between unplanted and planted surfaces was 21% (preceding) and 37% respectively. There were no significantly positive effects recorded during the summer periods [2].

Apart from that, common ivy has obvious benefits, as it can make a substantial contribution in improving the air quality, reducing noise pollution and increasing mental wellbeing.

This work hopes to simulate the growth pattern of ivy convincingly while considering the morphology foremost. The basis of this work is the programming language XL, for which this thesis will give a brief overview of functionalities and underlying language structure, and the interface software GroIMP.

1.1 Preview

For this work, the methods and focal points were largely predefined, the simulation of Hedera helix is multifaceted, the difficulties can be divided into three areas [3]:

1. Complexity and variability of plant growth,

2. Methodology of the representation of growth,

3. Translation into a computer language.

Complexity and variability refers to the growth characteristics that are to be represented in this work, including, for example, the number of internodes as well as the branching frequency, which can vary. Data collection, possible errors and inaccuracies will be discussed. Furthermore, environmental factors such as the weather or natural enemies in relation to the plant growth will be touched on.

The methology of representation and the resulting translation into a computer language though already implemented will be grazed, as a result formal languages are to be described and the L-systems introduced as being a core attribute of the XL programming language.

The first section of this thesis will describe the methods employed, for the most part describing the focal point, namely the L-Systems and their translation into a computer language. Then, a compact analysis section will handle the analysis of the plant structure, concluding this, a translation of the structure of the plant into the programming language will give a complete basis for the then following implementation.

The results show that the constructed plant aims at covering the structure in a similar way to ivy, it was shown that this plant can cover surfaces, in part, up to at least 88%. It is significant that the model can be changed to account for different growth forms and therefore isn't predetermined to these results.

Chapter 2

Methods and representation

As discussed in the previous chapter, there are three major difficulties this thesis has to contend, the topic of this first part will be the methods utilised to understand the growth patterns of ivy, translate them into a formal grammar using L-Systems and represent them using the XL-programming language.

2.1 Plant patterns and radiation

A thorough understanding of the plant will be a crucial part of this model, the predominant problem was that of finding reliable data and deciding which sources to use. Missing information will be declared as such in the model or be collected manually. For the sake of this model information pertaining to the internode length, the petiole-internode angle, petiole-leaf angle and the terminal growth was gathered[1]. The leaf texture was scanned and included.

The limited time and resources meant that, although mostly mitigated, inferences had to be made regarding the growth pattern. This included the analysis of the data collected for this work, these were analysed using python and common libraries (NumPy, SciPy, os, pandas, math). The light interception of leaves in the model marks an additional difficulty, to get reliable data the German Weather Service (DWD), a public authority, provides data on various weather phenomena, among these are datasets containing direct and diffuse light power accumulated ($\frac{\mathrm{kWh}}{\mathrm{m}^2}$) over a month on a horizontal plane. The collection partitions the land map of Germany into a grid of 1km $\times$ 1km cells, each of which either having -999 as an empty placeholder for which no information is offered, or a value corresponding to the light power in that area. The data that was of importance was that of the winter and summer months meaning December to February and June to August, to get a more reliable result the data of the last three years was used to calculate the mean light power of all datapoints in a month and of all three months, using this I now calculated an estimate

[1]with the help of Prof. Dr. Wolfgang Rohe

of what our vertical wall would receive as follows:

$$p_{\text{total}} = \frac{p_{\text{direct}}}{2} + \frac{p_{\text{diffuse}}}{2}$$

The reasons for this calculation, repeating again that these are only estimates, in which our building side (for growing ivy) facing south is:

- for the direct light, due to the higher angle of the sun toward the building side, there will be a loss of light intercepted, which is estimated to be the half,
- for the diffuse light, light interception would usually come from any side, the building would only facilitate one side accounting for the half.

This will allow a reasonable light model, this would differentiate between a direct and a diffuse light source, encompassing the angle of the sun for the direct source. Beyond this the library NumPy was used to test the mean, standard deviation, minimum and maximum of the measured values.

2.2 Representational methods

There are numerous approaches to simulating the growth of a plant, the method that seemed most sensible for this thesis was the programming language XL, which is based on L-Systems, a formal language in the broadest sense. The use of L-Systems has many advantages, among these are the parallel rewriting system, which allows for a simple transition between productions, the grammar-based approach allows its user to define the steps in an intuitive way while having a solid theoretical footing. The use of L-Systems will enable the modular definition of certain behavioural/growth patterns of the plant which will then be applied in selected parts of the plant. Finding out which pattern to represent and finding the right approach to simulating the ivy will play an important role. This section will give a short introduction of L-Systems and then transition to the XL programming language.

Definition 2.1 (Definition of an Alphabet)**.** The alphabet Σ consists of a set of symbols, e.g. $\Sigma = \{a, b, c\}$. A word w can make up any concatenation of symbols, any single symbol and the empty word ε.

This basic definition enables its user to make up words and an alphabet in order to later define languages, which have the obvious attribute of being made up of words. To describe all words over the alphabet the Kleene star is used which is defined as follows:

Definition 2.2 (Definition of the Kleene star/plus)**.** The Kleene star * over an alphabet describes the set of all words over that alphabet, e.g. $\Sigma^* = \{\varepsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, \dots\}$, The Kleene plus $^+$ over the alphabet describes the set of all words over the alphabet without the empty word ε.

Furthermore a few operators will be defined for later use:
Definition 2.3 (Definition of the length and concatenation)**.** Length ($|\cdot|$), the amount of symbols in a word, $|\varepsilon| = 0$. Multiplication ($\cdot$), the concatenation of words, e.g. abc $\cdot$ ab $=$ abcab.

The definition of grammars are an important basis to understanding L-Systems, these will be introduced here, the definition is similar to that given in [4].
Definition 2.4 (Definition of Grammars)**.** The tuple (N, T, S, R) makes up a grammar, these are:

- a finite number of non-terminals N which make up a number of symbols,
- a finite number of terminal symbols T for which $T \cap N = \emptyset$,
- a starting symbol S for which $S \in T$,
- a finite number of rules R which that make up the tuple (a, b) which are present in $\{(N \cup T)^* \cup N \cup (N \cup T)^*\} \times \{(N \cup T)^*\}$;

The former defintions enable the following introduction of L-Systems.

2.2.1 L-Systems

Central to the programming language XL are the L-Systems first introduced by Astrid Lindenmayer, these describe a grammar with a parallel rewriting mechanism, which was first defined to illustrate the growth patterns of filamentous organisms. [5] The L-Systems are based on the triple $(\Sigma, \omega, \mathrm{P})$, which will be described in the following definition:
Definition 2.5 (Definition L-Systems)**.** Σ is the alphabet, $\omega \in \Sigma^+$ is the Axiom and P the productions, with $P : \Sigma \to \Sigma^*$, the identity production is implied whenever no image under P is specified for a symbol.

This finding shows that the L-Systems are grammars, this form of L-Systems are a context-free, deterministic language [6]. This definition will be clarified by a following language example:
Example 2.1 (Example L-System)**.** $\Sigma = \{a, b, c\}$, $\omega = a$, $\delta_1 : a \mapsto ab$, $\delta_2 : b \mapsto c$

derivation	outset	production
1. derivation	a	ab
2. derivation	ab	abc
3. derivation	abc	abcc

Table 2.1: The first productions of an L-System language

The advantage of using this language is clear, the derivations are simultaneous, which will be useful to explain analogous bahaviours in different areas of an organism e.g. the growth of branches of the same plant will have very similar or exactly the same patterns, when being translated into a representational language, meaning the interpretation of the symbols into shapes or directions to form and illustrate the growth patterns. Following this, a few other parts are needed, these are turtle interpretations and direction symbols which will be represented as $\{+, -\}$ e.g. a 90° rotation to the left/right. The previous examples do not account for any directional interpretation of rules, which are one of the most important attributes of representational grammars, which are given by turtle interpretations of L-Systems [6]:

Definition 2.6 (Turtle interpretation)**.** A turtle state is made up by the triple: (x, y, α, d) where x, y are cartesian coordinates and α is the heading angle, the step size d. Note: the resulting coordinates can be calculated as $x' = x + d \cdot \cos(\alpha)$, $y' = y \cdot \sin(\alpha)$.

The following definition will explain the Stack operators [,] [6]:

Definition 2.7 (Stack operations)**.** The Stack operations [,] are used in unison, and are comparable to the stack operations [push, and] pop, meaning putting the current turtle state to the stack, pop will add the last element of the stack to the current turtle state, without any connection being drawn in between.

For this, the aforementioned symbols may allow for a demonstration, the step size of b is $d_1 = 0$ and of a $d_2 = 1$:

Example 2.2 (Example Turte interpretation)**.** $\Sigma = \{a, b, +\}$, $\omega = \{b\}$ *and* $P : \Sigma \to \Sigma^*$, $P : b \mapsto$ *a+a+b*, $P : a \mapsto$ *aa*

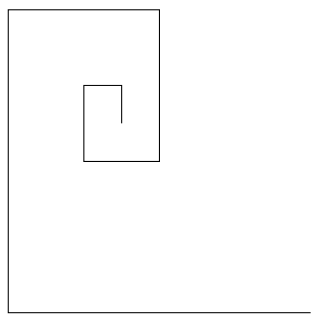

Figure 2.1: The fourth production of a graphical L-System language

For the depiction above, the symbol a will be represented by a line, giving an insight into how a representational grammar might be implemented, meaning a representation of the grammar beyond symbols, which will give a better understanding of how the programming language XL functions. Using L-Systems to simulate plant growth in this case meant that the exponential growth of these productions would have to be hampered, this was done by finding factors that impede the growth of the ivy and imitating these in the growth model.

2.2.2 Relational growth grammar

This section will firstly focus on relational growth grammars, which can be understood as a generalization of L-Systems to graphs, these are the underlying framework for the programming language. Following that specific parts of the XL programming language will be reviewed. The goal of being able to model the behaviour of objects using L-Systems demands information to be stored, called, deleted and restructured. More than that there need to be different shapes to represent the different parts and the solution should be efficient. These requirements can be addressed by a solution that is close at hand when considering the relationships between the parts, namely a graph structure.

The relations between the vertices of the graph are contained in the edges, the commonly used relationships will briefly be described:
Definition 2.8 (Common relations of edges)**.** The two relations between nodes are > successor and +> branch, the successor relation describes a following node, the branch relation successors inside of Stack operators.

This graph structure will contain the vertices or nodes that contain the information of the parts, and the edges which contain the relationships between the nodes of the graph e.g. the user will designate shapes to the different symbols of the L-System, all of these have the same parent class Node, the graph will be the result of a translation of the L-Systems into nodes and edges, the following example will clarify the relations between nodes:
Example 2.3 (Example relations between nodes)**.** *The derivation* $a[bc]d$ *will lead to the following relations:*

a+>b	b>c	a>d

This structure will enable the use of queries, which will help in locating nodes of the graph. The thesis by Ole Kniemeyer [7] will provide its reader with a context into how the graph is structured and the graph rewriting mechanisms.

2.2.3 The XL programming language

The programming language XL is based on Java, but does have many key differences, these will be discussed here.

L-Systems and common XL methods

The visual representation of L-System productions requires specifying the shapes that represent the symbols and setting out the rules, the shapes are set out by the language and can be extended in the keyword `module`. The production rules have three representations, for imperative rules

`::>`, for a replacement rule ==> and the SPO form ==» which cuts the edges between the host graph and the replaced graph, e.g. the replacement of `B` to `A` of the derivation `ABC` would result in a graph consisting of three disconnected nodes `A`, `A`, `C` where only the first is visible. There are positional operations which also extend the Node class, these include `RL`, `RH` and `RU` meaning the up, left and heading angles, referring to the right-hand rule. Gravitropism, meaning the force of gravity effecting the resulting representation, can be adjusted using `RV`, while `Translate(x,y,z)` will move the successive parts, in the global coordinate system. The following example is the implementation of the previously discussed turtle interpretation:

```
module B() extends Sphere(0.1){{
}}
module A() extends Cylinder(5, 3){{
        super.setColor(0);
}}
module plus() extends RU(90){{
}}
protected void init ()
[
        Axiom ==> B();
]

public void run ()
[
        B() ==> A() plus() A() plus() B();
        A() ==> A() A();
]
```

Listing 2.1: Implementation of the Turtle state 2.1

In the first part the classes are defined, which are synonymous to those of the previous example 2.1. The color of the `module` A is set to black. The next part method `init` includes the Axiom of the productions, this method gets called before the next derivations can take place. The `run` method will be called to perform the derivation.

There are many useful methods which are part of the Library class, these can be applied to Nodes. These include `getVolume(Node n)` which allows its user to get the volume of the node, `distance(Node a, Node b)` computes the distance of two nodes, among others, the Distributions class contains relevant stochastic distributions.

Queries and aggregation

The access to the nodes might be important when trying to specify the response of certain nodes to other ones or when gaining insight into variables. This can be done with help of queries, which rely upon the structure of the graph to search for patterns. The queries of the XL language can be defined by the user, frequently used relations are those previously described 2.8, following this the particular relations successor and branch will be reviewed.

Relations can be user defined and may include a directed (left, right) or undirected relation of nodes. A query is contained in the symbols $(^*, ^*)$ and may include conditions and relations, the parts are separated by comma. Types that are previously defined and can be accessed by a variable and the number of applications of a relation can be specified using * to indicate 0 to n repetitions, $^+$ to indicate 1 to n repetitions, $\{n, m\}$ n to m repetitions, $\{, m\}$ zero to m repetitions, $\{n, \}$ n to i repetitions, an exact number of repititions can be achieved by using $\{n\}$, these can be implemented as follows:

```
(*l:Leaf, l -ancestor-> Internode*)
(*i:Internode, l:Leaf, (distance(i,l)<0.5)*)
(*b:Bud, Internode +> (>)* b*)
(*p:Petiole, (getVolume(p)>0.02)*)
(*l:Leaf, i:Internode, i -descendants-> l*)
```

Starting from the first query, this would result in the ancestor internode of every leaf, the next describes all internodes and leaves that are in a distance of 0.5 of each other. Following the Internodes that have a branch node and any amount of successor nodes (0 to n) between them, all petioles that have a volume greater than 0.02, and all descendant internodes of all leaves. The variables can be accessed in a for loop. Another helpful tool of the XL programming language are the aggregation methods e.g. sum, mean, count among others, shorthand operators that tally up information on types easily, these can be implemented as follows:

```
count((*Bud*))
sum((*l:Leaf, (l.age>5)*).light)
mean((*Internode*).length)
```

These examples will count all the buds, sum up the light interception of all leaves age greater than 5 and take the mean of the length of all internodes. Queries and aggregation were both used in the implementation and were indispensable, they were necessary to select certain nodes, as a shorthand operation or to find nodes with specific attributes.

Chapter 3

Analysis

This chapter will provide the reader with an insight into the growth patterns of common ivy, while transitioning into how this model was implemented.

3.1 Hedera helix - a summary

The analysis of plant growth is a difficult undertaking, not only because of a lack of empirical data, but also of the many factors which must be taken into consideration making the simulation of ivy seem like a herculean task.

Ideas for modelling the plant growth had to be evaluated and, in some cases, discarded because of being untenable, of runtime considerations or simply having a too time-consuming application. Considering the numerous outside influences, e.g. sunlight, wind, rain, which in time give the plant its unique form, it did not seem feasible for the scale of this work, considering that the growth phases of Hedera helix are by themselves unique and complex.

3.1.1 Evaluation of data

As already discussed, this thesis includes data which was gathered manually, this includes the data gathered to better understand the plant morphology of ivy and the data which was later used to calculate the light interception.

Growth data

Firstly, the data describing the growth of the plant, which was collected by Prof. Dr. Rohe, will be listed. Starting with the internode length, for which multiple samples were taken from the main stem (n=14) twice in different positions and a side branch (n=10), these were analysed using common methods calculating the mean and standard deviation, a convergent internode length is presupposed, the distribution seems to complement that notion, using the Shapiro-Wilk test the normality of all three datasets was tested with non-significant ($\alpha = 0.05$) results for one of the main stems and the other of the side branch, the still ongoing growth might be used to explain the significant outcome. The model uses the results of the main stem.

Going on from that the petiole-internode angle (n=9) and the petiole-leaf angle (n=6) was established, the petiole-internode angle seemed to converge to $45°$ from the point of first appearance. The petiole-leaf angle is as it seems very variable and is influenced by the leaves moving toward the relative light optimal position, for which an algorithm was implemented, the mean positioning will be used as the initial value. Going on from that, the terminal growth will be included as a mutable variable, the datapoints supplement the data of a secondary source [8].

Light data

To calculate the light interception of the leaves reliably a big dataset would be necessary, our model does not take into account the possible conversion rate and only intercepts on the upper side of the leaf. The numerous constraints make it even more important to have reliable and expressive information which includes a longer time period and has multiple points of reference.

The process of finding a solution to this problem first led me to a local weather station for which Andreas Vohl presented me the radiation power measured in $(\frac{\mathrm{Wh}}{\mathrm{m}^2})$ over the course of two days, thanking him for his effort, I went on to search for a more comprehensive source for which I found the official site of the German Weather Service. This dataset partitions Germany into a grid of $1\mathrm{km} \times 1\mathrm{km}$ cells and gives the direct and diffuse lightinterception in a monthly basis in the unit: $(\frac{\mathrm{kWh}}{\mathrm{m}^2})$.

The conversions into this use case were already discussed, both datasets have an inaccuracy of $\pm 5\%$ over the summer and an inaccuracy of $\pm 15\%$ over the winter months, the origin of this being ground measurements and satellite data. [9, 10] The sum of the datapoints and then the mean was taken twice to get an overall monthly average light radiation.

For the implementation a parallelogram was put into frame in the GroIMP interface, the value for light interception for this big surface was then calculated, with the density, using the following formula:
$\mathrm{d_{exp}}$, $\mathrm{d_{rec}}$, expected and recorded radiation density, p_{curr} the current light density and the newly calculated light density to be applied to the model p_{new}:

$$\mathrm{p}_{new} = \frac{\mathrm{d_{exp}}}{\mathrm{d_{rec}}} \cdot \mathrm{p}_{curr}$$

the needed light density variable could be established.

The data that was analysed stretched over the three most recent years, one set was over the course of the winter months (December, January, February) the other over the summer months (June, July,

August) which lead to the following results, diffuse and direct light as the mean over the course of a month:
Summer direct: 48.26, summer diffuse: 37.07, winter direct: 4.53, winter diffuse: 8.63 (kWh/m^2) which indicates that the direct light is stronger than the diffuse light in summer, the inverse is true in winter.

Plant attributes

The plant attributes which were analysed, were already designated earlier, to get a short overview these are the internode length, the petiole-internode angle and the petiole-leaf angle, the terminal growth was also recorded but won't be used.

The internode length included (n=10) adjacent branch internodes and (n=28) main branch internodes, the expected behaviour of the plant is to grow logistically towards a value. Expecting the main internodes to have grown mostly to their convergent values the mean $\bar{x} = 4.44cm$ and standard deviation $\sigma = 1.14cm$ will be emulated in the model. The width of the internodes of the secondary vines will increase by half of that of the primary vine, due to prevalent growth of the primary vine, which was recorded earlier.

Calculation of the light density

The calculation of the light density was already addressed earlier in 3.1.1, the files included by the German weather service (DWD) both include the direct and indirect radiation on a monthly basis in a square meter. Opening and processing the files resulted in the expected results, this information was then transformed into the model with the DirectionalLight class light source in GroIMP.

3.1.2 Literature on ivy

Noticeably the aforementioned knowledge is not enough to describe ivy in any scope, for this, various literature sources had to be gathered and evaluated.

Growth patterns

An important source for understanding the growth patterns of ivy is primarily derived from two papers of Robbins 1957/1964. [8, 11] These describe two growth phases of ivy, the juvenile is described as having an alternating phyllotaxy, the leaves are described as having a three-pronged shape and stem and branches having aerial roots.

The juvenile stage can, as the paper indicates, be estimated to be ten years. The arborescent or adult ivy is characterised differently, its phyllotaxy is 2/5th meaning the next petiole having an angle of $144°$ in comparison to the previous. In this phase the branches grow without roots, the leaf form changes to a rounded form with one tip. The branches are described as growing orthotropic, meaning upright toward the source of stimulus, in this case meaning the sun. [12] Another important attribute of ivy for this work is the apical dominance of the plant, meaning the preferential growth of the main bud, this as a behaviour explains the former characterisation of the plant as being orthotropic, and is explained by the plant hormone auxin. [12]

Focusing on the branching of the ivy, the plant aims at covering the surface fully. This inherent pursuit of the plant will influence the branching order. Conceptually the branching order can be understood as the number of branches which have to be traversed from the current branch to the stem, as depicted below. The behaviour of filling a surface implies a high branching order.

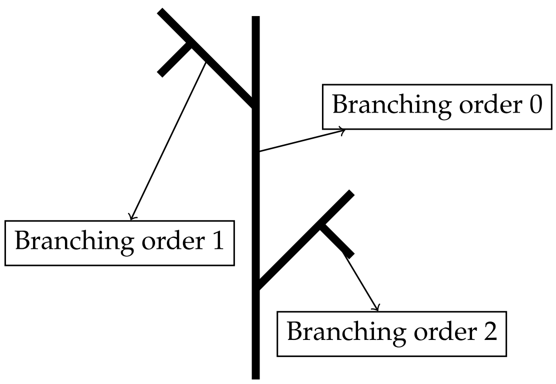

Figure 3.1: The branching order

Morphology

A simulation of a plant can only be as accurate as the information that it relies upon, this being the driving force a lot of research had to be done. A paper published by Daniele Castagneri, Matteo Garbarino and Paola Nola [13] gives a further insight into the density of ivy in a forest setting and the usual parameters the plant fulfils, salient for this task was the yearly radial growth rates, this was recorded for the span over last ten years and amounted to 1.03mm.

Another important source was that of Simona Rosu and Florin Sala [14] which described the expected length of an ivy as being up to $20-30$m, whereas the length of the leaves is described as being between $5-10$cm, the exact calculated means of the 50 samples were $\bar{l} = 8.96, \bar{w} = 10.50$ with the width to length ratio $r = 1.17$, the ratio is used to infer the width from the length.

Miscellaneous

The growing mechanism inherent to L-systems and by extension the XL programming language is the growth rate of the length of each word, which is as previously described for this example exponential. This does model the behaviour of plant growth well in its early stages, but the value of this estimation loses its explanatory power quickly for the remaining phases.

This problem is pronounced and very substantial and will have to be addressed. There are an abundant number of approaches that may be applicable to counter this problem, notably one of those considering the volume of the plant and calculating the additional growth for every time period, there are numerous different growth functions which seek to model growth in plants specifically.

The relative growth rate (RGR), which can be defined as the change of the total dry mass with respect to the time period and the initial mass, of Hedera helix is presented in a paper by Lawren Sack and Peter J. Grubb [15], which wasn't used as means for this simulation. Pivotal to this decision were, among other things, time constraints, but also the decision what function to use and the knowledge that the relative growth rate is not a constant [16] in the plants life span.

3.2 Translation into a programming language

Simulating the ivy with help of L-Systems requires the plant to be partitioned into smaller parts. Beyond only depicting the segments, they should represent an ivy in a way that would respect the technical description of the field and enable the user to intuitively understand the technical terms. The parts will have to either have a production in the L-System or terminate.

The simulation can either be handled by having an end state in mind and modelling all variables after that end state, which might have the advantage of being more accurate and efficient, but lacks the realism and rigour this model seeks out. The approach that seemed most palatable was that of set intervals at which the plant would grow and be updated. The figure below 3.2 will be

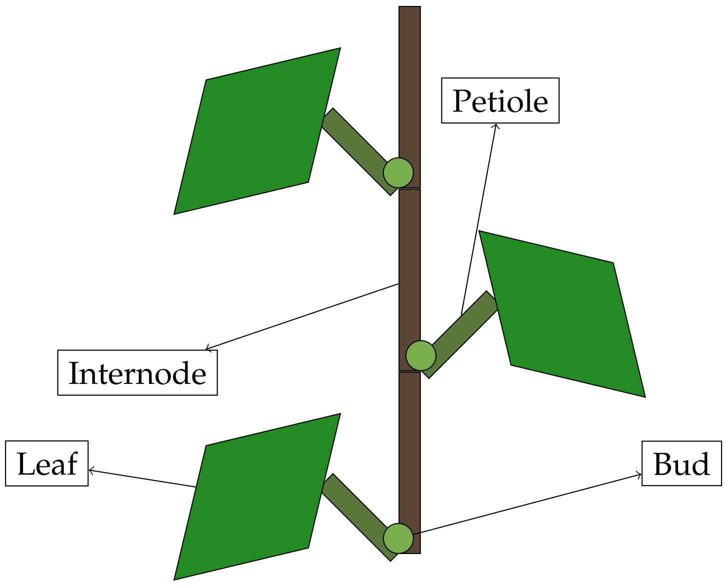

Figure 3.2: A representation of the plant structure

supportive in understanding the parts in which the plant was divided.

The growth will happen at the bud of the plant, meaning the simulation itself will start with the bud at a certain position, this bud will be active and will start growing, with each invocation of the growth methods the plant will grow by one internode, leaf, (dormant) bud and a petiole, the dormant buds themselves could get activated and start growing, the chance of that happening will be dependent on a logistic function. This is done to mitigate the problem of the plant growing uncontrollably.

The growth methods `juvenile` and `adult` refer to the terminal growth of the plant in the ivy growth stages, while the method `update` adjusts the parts of the plant to the growth experienced in that time interval. The time was, for the first part of the simulation, disregarded and came into play when looking at the end result. The terminal growth for each growth spurt is five applications of the `juvenile/adult` methods, which would see a growth of 22.2cm comparing this to the recorded data (n=2) which was taken over the course of one month in the summer months with a mean of 30.4cm seems like a plausible estimation of what monthly (or 1.5 months of) growth might be.
The time was calibrated using the fully grown juvenile plant, which should approximate about ten years [8].

Chapter 4

Implementation and results

The implementation is arguably the most important part of this thesis, the understanding of task did change, but the underlying goal remained, in convincingly representing Hedera helix both in its growth stages and in the resulting plant.

In this chapter, a short summary of the evaluation of the data, which was gathered for this work, will be given. Following that the practical implementation will be discussed, which will include the most pressing issues which were encountered, and how these were solved.

4.1 Technical implementation

The implementation included formulating different ideas, choosing the best approach and adding them to the model. It was important to keep the code as uncluttered as possible, which for the most part meant keeping it simple. To get a comprehensive overview of the model, different parts will be dealt with, the first part seeks to impart a better understanding of the plant structure, meaning the shapes used and the developing form of the ivy. Subsequently the light mechanics, meaning the light model used and the leaf repositioning will be considered, following that the last part will shed light on the growth patterns.

4.1.1 Plant structure

A glimpse of how the ivy is made up was already a topic in the previous chapter 3.2, this commonly used makeup of the plant is mirrored in the plant model. The ivy is dissected into the parts: internode, petiole, leaf, bud, which are represented by the shapes: frustum, cylinder, parallelogram and sphere (with a cone). The growth is modulated with help of a logistic function, the leaf, internode and petiole grow at the same pace relatively toward their expected values. New branches occur when a dormant bud is activated, this dormant bud then turns into a common bud and starts growing. The activation function is again a logistic one and is dependent on the age of the bud.

The example below is a practical implementation of the Petiole, this extends to the Cylinder class and adds further variables which are important later on, the setShader method sets the texture of the Petiole.

```
module Petiole(super.length, super.radius, int altern, int age, int order) extends
    Cylinder(length, radius){
  {
    super.setShader(EGA_6);
  }
};
```

Listing 4.1: Example of the module implementation

4.1.2 Light mechanics

GroIMP facilitates the use of different light sources, to emulate a realistic surrounding a light instance of the class DirectionalLight is used. In unison with the class LightModel, the number of rays and the light density was set. The method of how the light density was calibrated in relation to the gathered data is given in 3.1.1. There are two light sources, the diffuse light will have an 90° angle in relation to the surface, the direct light angle can be set by the user in the z-axis, conceptually the wall is facing south for the purpose of this example.

To set up the direct light source, the following methods were used:

```
public LightModel lm = new LightModel(10000000, 3);
module directLamp extends DirectionalLight(){
        {
          super.setPowerDensity(10000);     // Set power density;
          super.setRaylength(70);             // Set ray length;
          super.setVisualize(false);          // Visualize rays;
        }
};
module directLight extends LightNode{
        {
          super.setLight(new directLamp());   // Add lamp to light;
        }
};
```

Listing 4.2: Example of the implementation of the light source

The light model is first set up with ten million light rays and a depth of three, the following descriptive methods are expressive enough. For this example, the intercepted light for every patch will be calculated and added to the patch variable `patchlight`:

```
p:Patch(patchlight) ::> {
                        lm.compute();
                        p.patchlight = lm.getAbsorbedPower3d(p).integrate();
                }
```

Listing 4.3: Example calculation of the light intercepted by a patch

Broadly speaking, the leaves of a plant *move towards the relative optimal position with respect to their light interception*, this repositioning should be reflected in the simulation. This will be done by leveraging the Monte Carlo method to find the light optimal position. The method `optimize` will serve for this purpose. The method consists of a random number being generated, this number is part of a normal distribution for which the standard deviation is four and the mean zero. There are six cases for the three angles of the petiole and the leaf which have to be calibrated, starting out the light that is intercepted is calculated, then all the angles get changed by the previously calculated random number, the new light interception is calculated and compared to the previous one, the angle is changed if it leads to a higher numeric outcome in terms of the light intercepted.

4.1.3 Growth patterns

The growth patterns of the plant are majorly described by three characteristics which are mimicked in the plant model, for one the habit of the branches not to intersect, the effect of auxin among other hormones on the plants growth as described in 3.1.2, and the transition from the juvenile to the adult phase.

```
public void growTime(){
        for(1:this.timerange){
                this.time++;
                if(this.time<140 && !adulttransition()) for(apply(5)) juvenile();
                else for(apply(5)) adult();
                for(apply(1)) update();

                if(this.leafrepositioning) optimize();

                if(this.charts){
                        ...
                }
```

```
                this.volume.add(calculatevolume());
        }
}
```

Listing 4.4: The main method `growtime`

The most important functions are part of the `growtime` method, that is called to produce the plant, other methods are referenced later on.

The first characteristic was implemented for the buds which will, after each growth spurt, check their surroundings for new neighbouring internodes of the buds, if all the internodes in the neighbouring area cannot be accounted for as being of the same branch or of an area of the ancestor branch the bud will be deactivated, this will keep parts of the plant from growing uncontrollably and will lead to a plant pattern that is expected for ivy.

To simulate the effect of auxin on the plant, the growth of branches of order higher than zero is at first dampened, this effect linearly recedes until the branch is only affected slightly, the length at which the effect recedes is 23 internodes.

Though there is a reference value of about ten years 3.1.2 at which the ivy will transition from the juvenile to the adult phase, the reality is more faceted, this is why there needed to be an indicator for the transition which would fit the previously gathered data. This meant that the juvenile phase would have to end either if the ivy has the maximal length of about 30m, or if the growth of the ivy is impaired e.g. by competitor ivies. So there needed to be a variable that considers the plants relative size, this meant that if the additional growth of the plant, measured in volume, was less than a fraction of the maximum additional growth in a period, that the plant would transition into the adult phase.

4.1.4 Summary of the functionalities

There are various functionalities to the model, these are supposed to enable its user to easily change core attributes of the plant. The variable `branchgrowth` indicates how high the probability of dormant buds converting to active buds is, practically this would influence the number of branches growing. The variable `leafrepositioning` is a boolean that will move the leaves to a more light optimal position during runtime, if true. The charts variable will display multiple charts: the volume of the plant in relation to the time counter, the total light intercepted by the leaves in relation to time and the light intercepted by the leaves in relation to the other surfaces. The `lightangle` variable will influence the angle at which the direct light is pointing toward the wall, in the z-axis, and repositioning the light source to account for the discrepancy. The variable `timerange` indicates how often the variable `growtime` gets used in a time. Furthermore, the light density can be changed to account for different solar radiance, the negative gravitropism

(`RV`) is used for the branches to account for their heliotropism, meaning the plant moving its branches/leaves toward the light source, which can also be altered in the code.

4.2 Results

There are multiple results, for one there is the plant structure which, as seen in the previous section, is very malleable and is a product of the number of applications of the `growtime` method, then there are the measured values, which are also dependent on the plant that was produced.

For the sake of this example two plants were grown with the values: `branchgrowth = 0.6`, `leafrepositioning = true`, `timerange = 80`, the first during the winter months (December, January, February), with a direct light angle of `lightangle = 20` (degrees), and the second during the summer months (June, July, August) with `lightangle = 60` (degrees), notice that the position of the wall is supposed to face south, meaning the angle would be the zenith of the sun. Below the two examples are depicted 4.1.

(a) Winter lighting

(b) Summer lighting

Figure 4.1: Two ivies with different lighting

There are a few things to notice when looking at these images, the plants are of same height, even if it might not seem like it. Secondly, the leaves look different, this is because the leaves were repositioned toward a more light optimal angle, meaning for the winter plant the leaves are looking more towards the perspective, making for a visibly greener ivy, this might also change if the orientation of the plant is changed, this might lead to more diffuse light being intercepted by the plant. Another thing to note is that this plant doesn't have any neighbours which would fill out the space better and intertwine at points, limiting the horizontal growth of the ivy. Viewing the images also draws attention to the three grey squares behind the plant, these play a role in the subsequent part of calculating the light intercepted by these objects and make up a space of $1m^2$.

Of interest for this thesis is the development of the light intercepted by the three small grey squares behind the plant, this was done for the plant at hand. The results of the light intercepted, in kWh/m^2 over the course of a month, by the measurement points p for the time counter 1 was: $\overline{p_1} = 85.42226$ and, for the time counter 80, was: $\overline{p_{80}} = 15.10559$ which accounts for an 82.31% decrease, importantly the plant was still growing and might have a higher coverage if variables were different. The results for the winter plant, based on the light in kWh/m^2 over the course of a month, were: $\overline{p_1} = 13.17530$ and afterwards $\overline{p_{80}} = 3.19647$ which accounts for a 75.26% decrease.

(a) Winter lighting

(b) Summer lighting

Figure 4.2: Two ivies with different lighting, second plant

Another example might shed some more light on the variables used, the following example is again based on the summer and winter seasons, the `timerange = 80` again, this plant has a higher heliotropism and a `branchgrowth = 0.8` which leads to more dormant buds converting to growing buds, which lead to the result below 4.2. This plant got results of 77.39% light interception by the plant during the winter months and 88.85% during the summer months, which is an about $\sim 5\%$ increase in comparison to the previous example, notice that the plant isn't yet at the end of its growth period, this will take place up to 60 time ticks later, which would lead to a more dense growth pattern.

The data which was created by the previous plants was recorded and used to create two expressive plots, these show what the volume change of the plants were over the growth periods and the coverage of the patches previously discussed. The same difference in light coverage between the summer and winter seasons, the ratio of the coverage of the patches is also slowly flattening off, the adult phase with the protruding branches might again lead to a positive change in the coverage. To achieve a higher coverage the heliotropism might have to be higher, which would lead to less area having to be covered by the plant, another point that would lead to a higher coverage rate might be the tactile property which might have to be less pronounced. The expected result of both plants having very similar volume changes over the time steps did not occur.

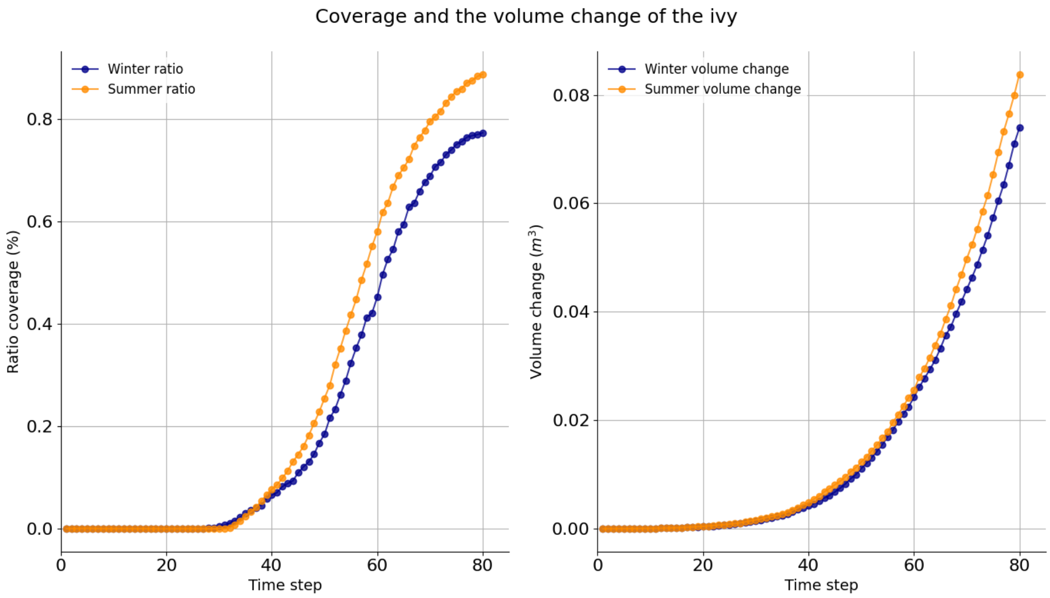

Figure 4.3: The coverage of the patches and the volume change

Previously in 3.1.1, the different light factors were of importance, these indicate that the diffuse light is the more prevalent light source during the winter months, while the direct light source is more pronounced in the summer months. The differences between the covering rates of the ivies might be explained by the higher angle of the direct light during the summer months, which leads to leaves throwing a bigger shadow, this might be ameliorated by the leaf repositioning method. Not taken into account are the wavelengths of the direct and the diffuse light which could, to a certain extent, be absorbed by the plant.

Chapter 5

Discussion

In this chapter the closing remarks and thoughts will be summarised, the literature review will give a more profound view of the literature that was gathered and processed. The prospects will include some of the findings and an outlook at what this work might entail. The conclusion will seek to give a short summary.

5.1 Literature review

The simulation of ivy proved to be a difficult endeavour, there were a few points which hindered the progress of the simulation, one emphasized point was the question of how to subdue the growth of the ivy which was mostly done by relying on the tactile characteristic of the plant. Realising the growth of a plant is a difficult undertaking and was limited by the information at disposal, this meant that a lot of research was necessary before this project could be finalised. This section will take a closer look at the resources at hand.

There were two parts that were pronounced in this work, for one the simulation of plant life and the use case of ivy as facade greenery. The programming language XL successfully implemented a system that utilises the L-Systems, the foundational work of Ole Kniemeyer [7] is now improved upon by further research on this topic and the ongoing work on the GroIMP framework. The simulation of plant life also includes the specific information on plants and the theoretical framework to model the growth of plants in the first place, with this especially growth functions are meant. There are only few resources describing the growth of ivy, the work of Robbins [8, 11] being the most informative on the subject of plant patterns and growth behaviour of ivy. Other sources included the radial growth rate of the internodes [13] and the leaf size of an ivy [14]. There are different ways of describing the growth of the dry weight of a plant, two important ones being the absolute and the relative growth rate, these can be described by different functions [17], though the relative growth rate is often used, it isn't a constant over the life span of the plant [16].

As already established at the outset the particular use of ivy as facade greenery is of importance for this work, this was underlined by the paper published by a research group in Reading, UK [2], though this topic was not introduced into the scientific discussion by these researchers, the amount of information that was gathered makes a very compelling argument about ivy having noticeable impact on the climate inside of a building, as already pointed out by the authors, the results came from comparing a cuboid with and without the ivy plant as facade greenery, how well these scale comparing them to a real world example might be asked.

5.2 Prospects

The hope is that this work can act as a basis for others to build upon, this will be briefly touched upon. It is close at hand that this work might be extended to other lianas, this might be done to plants of this type which have a similar composition. Furthermore the work on the ivy might be modified to include a more general approach, for instance it might be interesting to have the ivy grow on any surface, this might be done by minimizing the distance of the ivy toward the surface at every step. Another approach might be more interested in outside effects on the growth of the ivy, natural enemies, shadows of other objects, effects of certain hormones, weather or light dependence on the growth of the plant might be of importance. Furthermore it would be interesting to find out the density of different parts of the plant, with help of the volume the weight of the total ivy could be calculated which would allow the user to plot the relative growth rate (RGR) and the absolute growth rate, these might be indicators of how realistic the plant growth is and could be compared to the relative growth rate previously mentioned 3.1.2. The methods and approaches might also be helpful to any sort of plant simulation, many of the questions raised might have a more wide-reaching application. Hopefully the concise introduction into the XL programming language piques the interest of the reader.

5.3 Conclusion

The programming language XL convincingly builds upon Java functionalities, the implemented L-Systems work similarly to those first introduced by Astrid Lindenmayer and allow for an intuitive use. Queries are easy to work with and allow for a quick graph traversal, making the case for a more broad use of the programming language. However, the difficult access to pertinent information and the noticeable difference to Java complicate a more extensive adoption.

This short project on the simulation of ivy concludes with the observations that the XL programming language can be successfully utilised to model the behaviours of plants like ivy, while not relying on any obscure information. Furthermore the core characteristic of this plant of covering surfaces was reproduced successfully.

Bibliography

[1] M. Romanello, M. Walawender, S.-C. Hsu, A. Moskeland, Y. Palmeiro-Silva, D. Scamman, Z. Ali, N. Ameli, D. Angelova, S. Ayeb-Karlsson, S. Basart, J. Beagley, P. J. Beggs, L. Blanco-Villafuerte, W. Cai, M. Callaghan, D. Campbell-Lendrum, J. D. Chambers, V. Chicmana-Zapata, L. Chu, T. J. Cross, K. R. van Daalen, C. Dalin, N. Dasandi, S. Dasgupta, M. Davies, R. Dubrow, M. J. Eckelman, J. D. Ford, C. Freyberg, O. Gasparyan, G. Gordon-Strachan, M. Grubb, S. H. Gunther, I. Hamilton, Y. Hang, R. Hänninen, S. Hartinger, K. He, J. Heidecke, J. J. Hess, L. Jamart, S. Jankin, H. Jatkar, O. Jay, I. Kelman, H. Kennard, G. Kiesewetter, P. Kinney, D. Kniveton, R. Kouznetsov, P. Lampard, J. K. W. Lee, B. Lemke, B. Li, Y. Liu, Z. Liu, A. Llabrés-Brustenga, M. Lott, R. Lowe, J. Martinez-Urtaza, M. Maslin, L. McAllister, C. McMichael, Z. Mi, J. Milner, K. Minor, J. Minx, N. Mohajeri, N. C. Momen, M. Moradi-Lakeh, K. Morrisey, S. Munzert, K. A. Murray, N. Obradovich, M. B. O'Hare, C. Oliveira, T. Oreszczyn, M. Otto, F. Owfi, O. L. Pearman, F. Pega, A. J. Perishing, A.-C. Pinho-Gomes, J. Ponmattam, M. Rabbaniha, J. Rickman, E. Robinson, J. Rocklöv, D. Rojas-Rueda, R. N. Salas, J. C. Semenza, J. D. Sherman, J. Shumake-Guillemot, P. Singh, H. Sjödin, J. Slater, M. Sofiev, C. Sorensen, M. Springmann, Z. Stalhandske, J. D. Stowell, M. Tabatabaei, J. Taylor, D. Tong, C. Tonne, M. Treskova, J. A. Trinanes, A. Uppstu, F. Wagner, L. Warnecke, H. Whitcombe, P. Xian, C. Zavaleta-Cortijo, C. Zhang, R. Zhang, S. Zhang, Y. Zhang, Q. Zhu, P. Gong, H. Montgomery, and A. Costello, "The 2024 report of the lancet countdown on health and climate change: facing record-breaking threats from delayed action," *The Lancet*, vol. 404, no. 10465, pp. 1847–1896, Nov 2024. [Online]. Available: https://doi.org/10.1016/S0140-6736(24)01822-1

[2] R. W. Cameron, J. Taylor, and M. Emmett, "A hedera green façade – energy performance and saving under different maritime-temperate, winter weather conditions," *Building and Environment*, vol. 92, pp. 111–121, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360132315001778

[3] C. Godin and H. Sinoquet, "Functional–structural plant modelling," *New Phytologist*, vol. 166, no. 3, pp. 705–708, 2005. [Online]. Available: https://nph.onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8137.2005.01445.x

[4] K. Erk and P. D. L. Priese, *Theoretische Informatik*. Berlin, Heidelberg: Springer Verlag, 2000.

[5] A. Lindenmayer, "Mathematical models for cellular interactions in development i. filaments with one-sided inputs," *Journal of Theoretical Biology*, vol. 18, no. 3, pp. 280–299, 1968. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0022519368900799

[6] P. Prusinkiewicz and A. Lindenmayer, *The algorithmic beauty of plants*. Berlin, Heidelberg: Springer-Verlag, 1990.

[7] O. Kniemeyer, "Design and implementation of a graph grammar based language for functional-structural plant modelling," Ph.D. dissertation, Brandenburgische Technische Universität Cottbus, 02 2009.

[8] W. J. Robbins, "Gibberellic acid and the reversal of adult hedera to a juvenile state," *American Journal of Botany*, vol. 44, no. 9, pp. 743–746, 1957. [Online]. Available: http://www.jstor.org/stable/2438395

[9] Deutscher Wetterdienst, CDC - Vertrieb Klima und Umwelt. (2018) Datensatzbeschreibung rasterdaten der monatssummen für die direkte strahlung auf die horizontale ebene für deutschland basierend auf boden- und satellitenmessungen. [Online]. Available: https://opendata.dwd.de/climate_environment/CDC/grids_germany/monthly/radiation_diffuse/BESCHREIBUNG_gridsgermany_monthly_radiation_diffuse_de.pdf

[10] ——. (2018) Datensatzbeschreibung rasterdaten der monatssummen für die diffuse strahlung auf die horizontale ebene für deutschland basierend auf boden- und satellitenmessungen. [Online]. Available: https://opendata.dwd.de/climate_environment/CDC/grids_germany/monthly/radiation_direct/BESCHREIBUNG_gridsgermany_monthly_radiation_direct_de.pdf

[11] W. J. Robbins, "Topophysis, a problem in somatic inheritance," *Proceedings of the American Philosophical Society*, vol. 108, no. 5, pp. 395–403, 1964. [Online]. Available: http://www.jstor.org/stable/985812

[12] J. Kadereit, C. Körner, P. Nick, and U. Sonnewald, *Strasburger, Lehrbuch der Pflanzenwissenschaften*. Springer, 01 2021.

[13] D. Castagneri, M. Garbarino, and P. Nola, "Host preference and growth patterns of ivy (Hedera helix L.) in a temperate alluvial forest," *Plant Ecology*, vol. 214, no. 1, pp. 1–9, Jan 2013. [Online]. Available: https://doi.org/10.1007/s11258-012-0130-5

[14] S. Rosu and F. Sala, "Characterization of the Hedera helix L. leaves geometry based on some dimentional parameters and calculated ratios," *Life science and sustainable development*, vol. 3, pp. 7–16, 12 2022.

[15] L. Sack and P. J. Grubb, "The combined impacts of deep shade and drought on the growth and biomass allocation of shade-tolerant woody seedlings," *Oecologia*, vol. 131, no. 2, pp. 175–185, 2002. [Online]. Available: http://www.jstor.org/stable/4223240

[16] M. Rees, C. Osborne, I. Woodward, S. Hulme, L. Turnbull, and S. Taylor, "Partitioning the components of relative growth rate: How important is plant size variation?" *The American naturalist*, vol. 176, pp. E152–61, 09 2010.

[17] A. Pommerening and A. Muszta, "Methods of modelling relative growth rate," *Forest Ecosystems*, vol. 2, 12 2015.