

L-Grammatik Pflaumenbaum

Auswirkung normalverteilter Winkel in den Turtle Anweisungen

Fabian Rohne

Gregor Stockmann

29. März 2018

Inhaltsverzeichnis

1	Einleitung	4
2	Module	5
2.1	L-Systeme	5
2.1.1	Struktur	5
2.1.2	L-Systeme ohne Speicher	5
2.1.3	L-Systeme mit Speicher	5
2.2	Random	6
2.3	Tkinter	6
2.4	os	6
2.5	Turtle	6
2.5.1	verwendete Befehle	7
3	Analyse	8
3.1	allgemeine Analyse	8
3.2	kurze Beschreibung der einzelnen Versuchsreihen	8
3.2.1	Winkel 5° mit unterschiedlichen Ebenenzahl	8
3.2.2	Winkel 10° mit unterschiedlichen Ebenenzahl	9
3.2.3	Winkel 15° mit unterschiedlichen Ebenenzahl	9
3.2.4	Winkel 5° mit halber Astlänge	9
4	Normalverteilung	10
4.1	Theoretischer Hintergrund	10
4.2	Versuchsaufbau	11
4.3	Vermutungen	11
4.4	Auswertung	13
4.4.1	Versuchsreihe 1	13
4.4.2	Versuchsreihe 2	13
4.4.3	Versuchsreihe 3	14
4.4.4	Versuchsreihe 4	15
4.5	Fazit	15
5	aufgetretene Probleme	16
5.1	Probleme mit dem L-System	16
5.2	Probleme mit Turtle	16
5.3	Probleme mit Python	16
6	Schlusswort oder das Geheimnis der schönen Bäume	17

Abbildungsverzeichnis

1	Gaussglockenkurve der Standardnormalverteilung	10
---	--	----

2	Normalverteilung um den Eigenwert 12	11
3	Normalverteilung um den Eigenwert 18	12
4	Versuchsreihe 1	13
5	Versuchsreihe 2	14
6	Versuchsreihe 3	14
7	Versuchsreihe 4	15

Tabellenverzeichnis

1	Verteilung der Winkel um den Eigenwert 12	12
2	Verteilung der Winkel um den Eigenwert 18	12

1 Einleitung

Gegenstand der Forschung ist ein von den Verfassern geschriebenes Python 3 Programm. Das Programm erzeugt mittels eines L-Systems einen String, welcher einen Pflaumenbaum modelliert. Um den String zu visualisieren werden die Substrings in Anweisungen für das Grafikmodul Turtle übersetzt.

Die Verfasser gehen in dem Projektbericht der Frage nach welchen Einfluss es auf die Gestalt des Baumes hat wenn die Richtungsanweisungen für die Turtle von verschiedenen Normalverteilungen abhängig sind. Die wahl viel auf die gauss'sche Normalverteilung, da sie in der Natur eine häufig auftretende Wahrscheinlichkeitsverteilung ist.

2 Module

2.1 L-Systeme

Das grundlegende Prinzip von L-Systeme besteht darin, einfache Elemente eines Objektes mithilfe von Produktionsregeln durch Kompliziertere zu ersetzen (Ableitung). Dabei können diese Regeln wieder auf zuvor ersetzte Bereiche angewendet werden. Die Anzahl der Ersetzungsschritte wird als Tiefe des L-Systems bezeichnet.

2.1.1 Struktur

Ein L-System setzt sich aus vier Elementen, dem Startsymbol, einer Regelmenge, einem Alphabet und einem Vokabular, zusammen.

Mit dem Startsymbol, auch Axiom, beginnt das "Ersetzungssystem". Dabei ist es essentiell, dass das Startsymbol ein Nichtterminalsymbol ist. Das heißt, ein Symbol, dass kein endgültiges Zeichen ist und wiederum abgeleitet werden kann.

Auf das Startsymbol wird die Regelmenge, bestehend aus Produktionsregeln, angewandt. Eine Produktionsregel ist ein zuvor definiertes geordnetes Paar (α, β) , auch $(\alpha \rightarrow \beta)$, welches die linke Seite, mit mindestens einem Nichtterminalsymbol, durch die rechte Seite ersetzt.

Diese Regeln setzen sich aus dem sogenannten Vokabular zusammen. Das besteht aus der disjunkten Vereinigung des Alphabetes und der Menge der Nichtterminalsymbolen. Das Alphabet umfasst die Menge aller Terminalsymbole.

2.1.2 L-Systeme ohne Speicher

L-Systeme ohne Speicher setzen sich aus einer simplen Aneinanderreihung von Befehlen zusammen, welche zuvor nur durch eine einzige Produktionsregel erzeugt wurde.

2.1.3 L-Systeme mit Speicher

Bei einem L-System mit Speicher wird zuvor ein entsprechendes Speichermedium, z.B. eine Liste oder Ähnliches, definiert. In dieses Speichermedium wird über ein Befehl, alle derzeitigen Eigenschaften, wie Ausrichtung, x- und y-Koordinaten des Zeichenmoduls übertragen. Nachdem ein von dort aus weitere Segmente gezeichnet wurden, können die Daten aus dem Speichermedium wieder abgerufen werden und es kann zu dieser Position zurückgekehrt werden.

2.2 Random

"Random" ist ein in Python 3 importierbares Modul zur Generierung von zufälligen Zahlen und Symbolen. Mit dem Befehl `"import random"` wird das Modul für den aktuelle Programmcode verfügbar.

Zur Erzeugung einer Zufallszahl zwischen 0 und 1 beinhaltet das Modul "Random" einen eigenen Befehl. `"random.random()"`. Diese Anweisung wird besonders Wahrbei der Verwendung von Wahrscheinlichkeiten hilfreich und nützlich.

2.3 Tkinter

Das Modul "Tk interface", kurz "Tkinter", ist zur Gestaltung von Labeln und Beschriftungen zu Texten und Bildern entwickelt worden. Dabei ist das Label nur zur Visualisierung und zu keiner Interaktion gedacht. Dazu wird das Modul mit den Befehlen `"import tkinter"` oder `"from tkinter import *"` importiert. In unserem Programmcode verwenden wir das `"Tkinter"` - Modul zur Beschriftung der erzeugten Vektorgrafiken.

2.4 os

Mit Hilfe des importierbaren Modules "os" kann eine Verbindung zwischen Betriebssystem und dem Pythoncode hergestellt werden. Dazu muss das Modul zuerst mit `"import os"` in das Programm eingefügt werden.

Mit Hilfe dieses Modules erstellen wir für jeden Datensatz einen eigenen Ordner, in welchen anschließend alle dazugehörigen Vektorgrafiken als `".eps"` - Dateien gespeichert werden.

2.5 Turtle

"Turtle" ist ein in Python 3 importierbares Modul zur Erstellung von Bildern und Grafiken. Dabei kann das Zeichenelement, die sogenannte "Turtle", mit sehr einfach Anweisungen gesteuert werden. Die Befehle, einzeln oder als Liste, werden nacheinander ausgeführt und stehen für jeweils eine Anweisung der Turtle. Des Weiteren können die Befehle mit Schleifen und range-Befehlen verkürzt werden. Außerdem besteht die Möglichkeit Funktionen zu schreiben. Mit Hilfe der oben genannten Möglichkeiten werden in unserem Programmcode die dekorativen Elemente (Blätter, Blüten, Pflaumen) generiert.

2.5.1 verwendete Befehle

Befehl	Auswirkungen
forward(distance)	bewegt die Turtle um die Entfernung "distance" vorwärts
left(angle)	dreht die Turtle um den Winkel "angle" nach links
right(angle)	dreht die Turtle um den Winkel "angle" nach rechts
setheading(angle)	dreht die Turtle auf den Winkel "angle". Dabei ist "angle" = 0 die Ostausrichtung
heading()	nimmt den derzeitigen Ausrichtungswinkel der Turtle auf
xcor()	nimmt die derzeitige x-Koordinate der Turtle auf
ycor()	nimmt die derzeitige y-Koordinate der Turtle auf
position()	gibt die x- und die y-Koordinate der Turtle aus
begin_fill()	beginnt das Ausfüllen
end_fill()	beendet das Ausfüllen
fillcolor('color')	füllt die Figur mit der Farbe "color" aus
speed(speed)	die Turtle zeichnet mit der Geschwindigkeit "speed"
up()	hebt den Zeichenstift ab
down()	senkt den Zeichenstift ab
goto(xcor(), ycor())	bewegt die Turtle zu dem Punkt mit der x-Koordinate und y-Koordinate
done()	beendet das Zeichnen
exitonclick()	schließt das Fenster mit einem Klick
bye()	schließt das Turtlefenster

3 Analyse

3.1 allgemeine Analyse

Auf die Entstehung des Baumes haben viele Faktoren Einfluss.

Besonders stark beeinflusst die Regelmenge die Struktur des Baumes. So entscheiden die Produktionsregeln, in Kombination mit ihren Wahrscheinlichkeiten, über das grundlegende Wachstumverhalten der Bäume.

Durch die Wahl der verwendeten Produktionsregeln, sind in den Bäumen sehr viele kahle Äste an zu finden. Diese treten vermehrt als Enden auf und fallen dem Betrachter somit sofort ins Auge.

Ein weiterer bedeutender Faktor ist die Standardabweichung. Diese hat neben der Regelmenge den größten Einfluss.

Bei kleinen Winkeln mit geringer Standardabweichung wachsen alle Bäume sehr gerade und gebündelt nach oben. Dabei treten verhältnismäßig wenige Überlappungen von Ästen und dekorativen Elementen auf. Des Weiteren findet eine harmonische Verteilung dieser Elemente statt.

Mit zunehmender Standardabweichung wachsen die Bäume nun breit gefächerten. Aufgrund der wachsenden Normalverteilung steigt die mögliche Abweichung der Winkel. Daraus resultiert das vermehrte Auftreten der Überlappungen.

Infolge zu großer Winkel und Standardabweichungen wachsen einzelne Äste und Verzweigungen zu stark nach unten und über den Beginn des Hauptstammes und des "Turtle"-Fensters hinaus. Sehr auffällig ist, die drei Strukturarten, welche die Bäume häufig annehmen.

Die erste Wachstumsart, ist die, dass der Baum sehr gebündelt und mit vielen Überlappungen in eine Richtung wächst. Dabei treten nur vereinzelt wesentlich kleinere Verästelungen auf, welche in besonders stark in andere Richtungen wachsen.

Bei einer Weiteren kommt es zu einer großen Auffächerung und Teilung des Baumes. Dabei wächst der Baum sehr stark in die Breite und es kommt zu einer Aufteilung von zwei oder mehr Sektionen, welche sich stark voneinander weg entwickeln.

Die letzte Struktur ist eine Kombination aus den oben erläuterten. Dabei findet als Erstes eine weite Aufteilung der Hauptäste statt. Anschließend wachsen davon ausgehend weitere sehr gebündelt und gerichtet Äste.

Sehr auffällig, besonders bei wenigen Verzweigungsebenen, sind die vielen kahlen Äste. Die treten vor allem an den Ausläufern der Bäume auf und sind somit sehr auffällig für den Betrachter.

3.2 kurze Beschreibung der einzelnen Versuchsreihen

3.2.1 Winkel 5° mit unterschiedlichen Ebenenzahl

Anfangs sorgt der geringe Winkel für sehr gerade und streng nach oben wachsende Bäume. Mit zunehmender Standardabweichung und Ebenenzahl werden die Bäume zu-

nehmend komplexer und wachsen unkontrollierter. Besondere Auffälligkeiten sind dabei z.B., dass bei einer Ebene zu bis einer Standardabweichung von 25 nur zwei kahle Äste entstehen. Darüber hinaus entwickeln sich zwei dekorative Elemente und kleinere Äste.

3.2.2 Winkel 10° mit unterschiedlichen Ebenenzahl

Anfänglich zählt sich der größere Winkel aus, da weniger Überlappungen von Ästen und dekorativen Elementen auftreten und sehen somit schöner aus.

Mit zunehmender Normalverteilung steigt die Anzahl der Überlappungen wieder stark an.

3.2.3 Winkel 15° mit unterschiedlichen Ebenenzahl

Zu Beginn mit einer geringen Standardabweichung findet aufgrund des größeren Winkels eine angenehmere und ausgeglichene Verteilung der dekorativen Elemente statt. Dadurch wirkt auch der Baum harmonischer. Sobald aber die Standardabweichung steigt, ist sofort ein unkontrolliertes und wildes Wachstum der Bäume zu beobachten.

3.2.4 Winkel 5° mit halber Astlänge

Aufgrund der halben Astlänge werden die Bäume sehr kompakt und klein. Dabei kommt es zu vielen Überlappungen. Aufgrund der Kompaktheit der Bäume fallen die kahlen Äste nicht mehr so stark ins Auge, wodurch der Baum harmonischer und schöner aussieht.

4 Normalverteilung

4.1 Theoretischer Hintergrund

Die Normalverteilung kann durch zwei Parameter eindeutig bestimmt werden. Dem Erwartungswert(μ) und der Standardabweichung(σ). Eingesetzt in die Formel für die Wahrscheinlichkeitsdichte(1) ergibt sich die charakteristische Gaussglockenkurve(Abb.1)

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \quad (1)$$

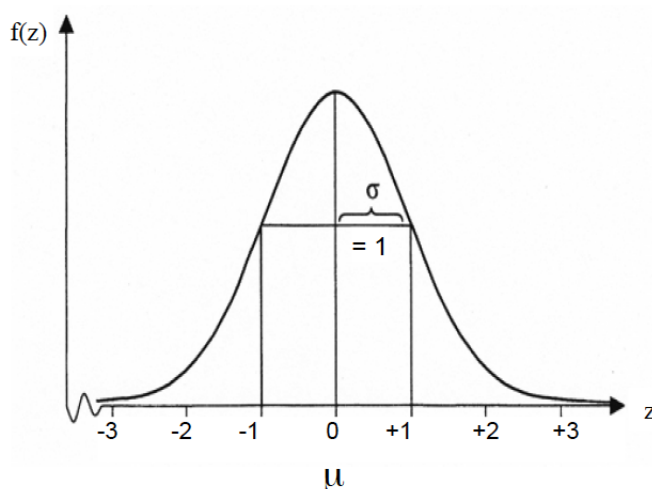


Abbildung 1: Gaussglockenkurve der Standardnormalverteilung

Die gauss'sche Glockenkurve ist symmetrisch zum Eigenwert, der ebenfalls das Maximum der Funktion ist. Eine Änderung des Eigenwertes führt zu einer Verschiebung entlang der x-Achse. Summe und Differenz aus Eigenwert und Standardabweichung ergeben die Werte an denen die Wendepunkte der Funktion zu finden sind. Das Intervall $[\mu - \sigma, \mu + \sigma]$ umfasst 68,27% der Fläche unter der Gaussglockenkurve. Im nächst größeren Intervall $[\mu - 2\sigma, \mu + 2\sigma]$ befinden sich 95,45%. Vergrößert man das Intervall $[\mu - 3\sigma, \mu + 3\sigma]$ erneut liegen 99,73% der Fläche darin. Während man also das Intervall gleichmäßig vergrößert wird der Flächenzugewinn immer kleiner jedoch nie Null, da die Funktion sowohl für negative als auch positive Werte gegen Null strebt. Die gesamte Fläche unter der Kurve ist 1, entspricht also einem sicheren Ergebnis. Ändert man die Standardabweichung folgt daraus eine Stauchung bzw. Streckung der Kurve.

4.2 Versuchsaufbau

Turtle bekommt für jedes '+' im Code die Anweisung sich nach Rechts zu drehen. Der Substring '-' löst eine Linksdrehung der Turtle aus. Die Größe der Winkel um die gedreht wird unterliegt einer Normalverteilung. Erwartungswert und Standardabweichung müssen in der Funktion 'main_gen' eingegeben werden. Für die ersten Versuche ist der Erwartungswert der Rechtsdrehungen 12° und der Linksdrehungen 18° . Die Erwartungswerte sind unterschiedlich um eine Symmetrie zu vermeiden. Die Standardabweichung ist bei beiden Drehungen gleich. In der ersten Versuchsreihe liegt sie bei 1° in der zweiten bei 2.5° , bei der dritten bei 5° und schließlich bei der vierten bei 10° . Pro Versuchsreihe wurden 100 Bäume generiert um eine empirische Aussage über den Einfluss verschiedener Standardabweichungen treffen zu können.

4.3 Vermutungen

Bei geringer Standardabweichung ist die Wahrscheinlichkeit sehr gering, dass besonders große oder kleine Winkel auftreten. Mit einer Sicherheit von 68,27% liegen die Winkel in zwischen 11° und 13° bzw. 17° und 19° . Winkel im Intervall $(7^\circ, 8^\circ)$ oder $(16^\circ, 17^\circ)$ bzw. $(13^\circ, 14^\circ)$ oder $(22^\circ, 23^\circ)$ kommen lediglich mit einer Sicherheit von unter 0,1 Prozent vor. Die Bäume aus dieser Serie werden damit ähnlich groß und weit ausgedehnt sein, da durch den Code festgelegt ist wie viele Ebenen sie haben. Die Tabelle lässt die Schlussfolgerung zu, je größer die Standardabweichung desto höher die Wahrscheinlichkeit das viel größere oder viel kleinere Winkel auftreten. Daraus folgt bei größerer Standardabweichung werden die Bäume der Versuchsreihe sich stark unterscheidende Größen und Ausdehnungen haben. Es kommt einfach gesagt zu extremeren Unterschieden im Aussehen. Bei der vierten Versuchsreihe besteht eine Sicherheit von 13,6% das sich die Winkel im Intervall $(-8^\circ, 2^\circ)$ bzw. im Intervall von $(-2^\circ, 8^\circ)$ demzufolge kann aus einer Linksdrehung eine Rechtsdrehung werden und umgekehrt.

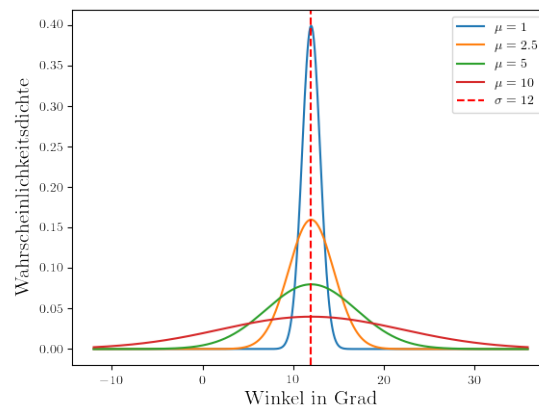


Abbildung 2: Normalverteilung um den Eigenwert 12

	σ in Grad	$[\mu - \sigma, \mu + \sigma]$	$[\mu - 2\sigma, \mu + 2\sigma]$	$[\mu - 3\sigma, \mu + 3\sigma]$
Anteil der Winkel in %		68,27%	95,45%	99,73%
Versuchsreihe 1	1	11;13	10;14	9;15
Versuchsreihe 2	2.5	9,5;14.5	7;17	4,5;19,5
Versuchsreihe 3	5	7;17	2;22	-3;27
Versuchsreihe 4	10	2;22	-8;32	-22;42

Tabelle 1: Verteilung der Winkel um den Eigenwert 12

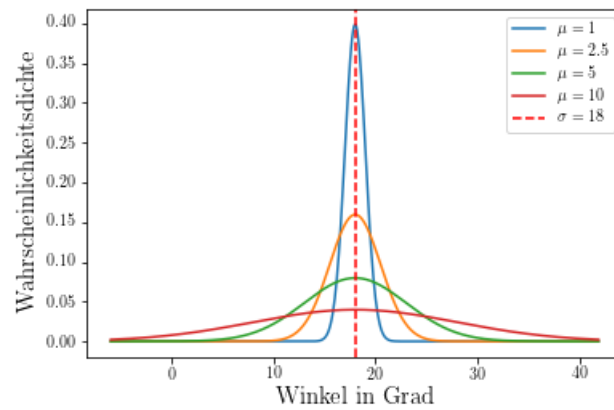


Abbildung 3: Normalverteilung um den Eigenwert 18

	σ in Grad	$[\mu - \sigma, \mu + \sigma]$	$[\mu - 2\sigma, \mu + 2\sigma]$	$[\mu - 3\sigma, \mu + 3\sigma]$
Anteil der Winkel in %		68,27%	95,45%	99,73%
Versuchsreihe 1	1	17;19	16;20	15;21
Versuchsreihe 2	2.5	15,5;20,5	13;23	10,5;25,5
Versuchsreihe 3	5	13;23	8;28	3;33
Versuchsreihe 4	10	8;28	-2;38	-12;48

Tabelle 2: Verteilung der Winkel um den Eigenwert 18

4.4 Auswertung

4.4.1 Versuchsreihe 1

Die Vermutungen haben sich Teilweise bestätigt alle Bäume haben eine ähnliche Höhe. Unterschiede gibt es lediglich bei der Ausdehnung hier können zwei Typen unterschieden werden. Der erste Typ wird durch den Baum 27 repräsentiert die Hauptäste wachsen auseinander. Im Gegensatz dazu steht der Baum 95 bei ihm wachsen die Hauptäste in einander. Allerdings ist die Ursache des Problems bei den Regeln im L-System zu finden. Denn der String gibt Auskunft ob die Äste nach rechts oder links wachsen.



(a) Baum 27 von 100

(b) Baum 95 von 100

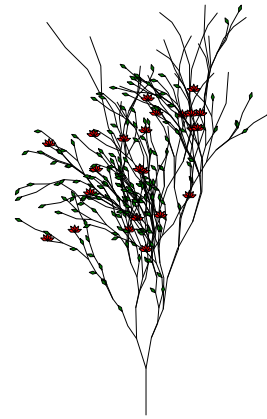
Abbildung 4: Versuchsreihe 1

4.4.2 Versuchsreihe 2

Die Vermutung, dass bei einer höheren Standardabweichung höhere und weiter ausgebreitete Bäume entstehen hat sich bestätigt. Es sind erneut die beiden Typen aufgetreten. Baum 63 ist weit ausgedehnt, Baum 97 dagegen eng zusammen gewachsen. Ein wesentlicher Unterschied ist das erneute Zusammenwachsen der Hauptäste in der Krone. Grund für das erneute Verwachsen ist die höhere Standardabweichungen, denn dadurch kommen vermehrt größere Winkel vor. Somit sind abruptere Richtungswechsel möglich.



(a) Baum 63 von 100



(b) Baum 97 von 100

Abbildung 5: Versuchsreihe 2

4.4.3 Versuchsreihe 3

Es kommt immer weniger zu starken Überlappungen. Der Baum ist stärker auf der Fläche verteilt. Denn durch die hohe Standardabweichung werden Tendenzen in bestimmte Richtungen durch die abwechselnd größeren und kleineren Winkel ausgeglichen. Trotzdem ist weiterhin eine grobe Einteilung in zwei Typen möglich: Die dicht gewachsenen Bäume zum Beispiel TBaum 35 und die locker gewachsenen wie Baum 50.



(a) Baum 35 von 100

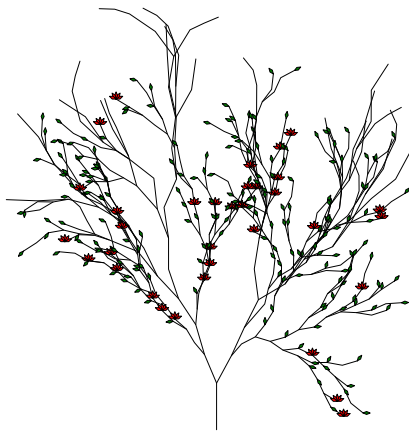


(b) Baum 50 von 100

Abbildung 6: Versuchsreihe 3

4.4.4 Versuchsreihe 4

Die Tendenzen aus Versuchsreihe 3 setzen sich weiter fort. Es gibt nun kaum noch Bäume die nicht mehr in die Breite wachsen. Diese Erkenntnis steht den Vermutungen entgegen, da durch das zunehmende auftreten kleinerer Winkel ebenfalls noch kompaktere Bäume erwartet wurden. Allerdings scheinen sich große und kleine Winkel weiterhin auszugleichen. Exemplarisch dafür sind die Beispielbäume in Abbildung 7. Baum 9 ist einer der kompaktesten Bäume der Versuchsreihe. Baum 73 ist am weitesten ausgedehnt.



(a) Baum 9 von 100



(b) Baum 73 von 100

Abbildung 7: Versuchsreihe 4

4.5 Fazit

Anders als in der Vermutung formuliert kommt es mit wachsender Standard Abweichung nicht zu einem größeren unterschied innerhalb der Versuchsreihe. Festzustellen ist, dass mit größerer Standardabweichung die Ausdehnung größer wird und es somit zu weniger Überlappungen kommt. Diese Entwicklung ist zum einen auf den Ausgleich der Bildungsregeln und zum anderen auf den Ausgleich großer Winkel durch kleine Winkel zurück zu führen.

5 aufgetretene Probleme

5.1 Probleme mit dem L-System

Viel Arbeit und Überlegungen mussten wir anfangs in das L-System stecken. Dabei war das grundlegende Konzept des Ersetzungssystems korrekt. Dennoch wurden die Anweisungen zum Zeichnen der Äste, durch weiterführende Produktionsregeln komplett ersetzt, was dazu führte, dass der Baum mehr Blüten und Blätter als Äste besaß. Zur Lösung dieses Problems haben wir, nachdem das entsprechende Element ersetzt wurde, dieses wieder davor gesetzt und als neuen String definiert.

5.2 Probleme mit Turtle

Ein anfängliches Problem mit Turtle war das Schließen des Zeichenfensters. Nachdem die Zeichnung fertiggestellt wurde, ließ sich das Zeichenfenster nicht problemlos beenden. Das Notebook wurde von den Kernels getrennt und benötigten einen kompletten Neustart.

Dieses Problem ließ sich durch den Turtle-Bye()-Befehl oder den Befehl `"exitonclick()"` zusammen mit der Verwendung von "Spyder" beheben.

Des Weiteren musste jede Zeichnung vom Anwender separat in Auftrag gegeben und abgespeichert werden. Da dieses Vorgehen sehr zeitintensiv ausfällt, ist somit nicht für eine umfassende Analyse mit vielen Untersuchungsobjekten geeignet ist, wurde der Prozess mit den Modulen "os" und "Tkinter" automatisiert.

5.3 Probleme mit Python

Eine weitere Komplikation bestand darin, dass Python trotz korrekter Einschübe, keine Mischung von Leerzeichen und Tabulatoren zulässt. Dies führt mehrmals dazu, dass der richtige Programmcode nicht funktionierte. Da sowohl "Spyder" als auch das "Jupyter Notebook" keine Möglichkeit anbieten Absätze, Leerzeichen und Tabulatoren anzuzeigen, mussten wir den gesamten Programmcode überprüfen.

6 Schlusswort oder das Geheimnis der schönen Bäume

Nach rund 1500 generierten und analysierten Bäumen können wir nun ungefähr abschätzen, wie schöne und erfolgreiche Bäume generieren werden können.

Da die Produktionsregel die grundlegende Baumstruktur vorgeben, sollten dort an erster Stelle viele Regeln neue Äste generieren. Hinzu kommt, dass nicht zu viele weitere dekorative Elemente eingebunden werden sollten, da der Baum sonst zu überladen wird. Des Weiteren um kahle Äste zu vermeiden sollte in fast jeder Regel am Ende ein Blatt generiert werden.

In Bezug auf die Winkel sollten kleine Winkel zw. 5° und maximal 15° mit einer Normalverteilung zwischen 5 und 15 verwendet werden. Dabei ist zu beachten das bei kleineren Winkel eine verhältnismäßig größere Normalverteilung benötigt wird und umgedreht.

Auch sollten mindestens 5 Ebenen und nicht mehr als 10 Ebenen generiert werden, da die Bäume sonst zu klein sind und zu viele Überlappungen auftreten bzw. der "Turtle" - Screen begrenzt und somit zu klein ist.

Aufgrund der vielen verwendeten Zufallsvariablen entstehen die Bäume nicht deterministisch. Somit ist die wichtigste Grundlage für schöner Bäume die Generierung von Bäume im großen Stil und Geduld.