

Zwischenbericht zum Projekt: Entwicklung
statistischer und geometrischer Verfahren zur
Charakterisierung von Pflanzen- und
Bestandesarchitekturen und für Modellvergleiche
(KU 847/3-1)

26. Mai 2003

Inhaltsverzeichnis

1	Einleitung	1
2	Lindenmayer-Systeme	2
2.1	GROGRA	2
2.2	Erweiterungen	3
3	GRODISC	7
3.1	Motivation und Aufgabe	7
3.2	Die Datenstruktur	8
3.3	Algorithmen	11
3.4	Funktoren	13
3.5	Datenschnittstellen	14
3.6	Benutzerschnittstelle	16
3.7	Diskussion	16
4	Rohdatenerhebung	18
4.1	Motivation	18
4.1.1	Die Untersuchungsfläche	19
4.1.2	Aufnahmeparameter und Messmethoden	19
4.2	Ergebnisse	20
4.2.1	Allgemeine Statistiken	20
4.2.2	Korrelationsanalysen	22
4.2.3	Analyse der Kronenabstände	23
5	Ein Bestandesmodell für Kiefer	24
5.1	Der Einzelbaum	24
5.2	Interaktion	25
5.3	Das Wachstum	27
5.4	Das L-System	28
5.5	Ergebnisse	30

5.6	Diskussion	30
6	Modellaggregation einer LIGNUM-Kiefer	32
6.1	Material und Methoden	32
6.2	Ergebnisse	34
6.3	Diskussion	35
7	Kooperation und Präsentation der Ergebnisse	37
	Literatur	40

1 Einleitung

Es existiert eine Vielzahl von Modellen, die sich mit dem Wachstum von Pflanzen im Allgemeinen bzw. Baumwachstum im Speziellen beschäftigen. Klassische forstliche Ansätze (Prodan (1965), Kramer (1984)) beziehen sich auf einzelne Variablen von Individuen bzw. Baumbeständen wie die Höhe oder den BHD bzw. deren Verteilungen in einem Bestand. Diese Ansätze haben neben anderen den Nachteil, Konkurrenz zwischen Individuen zu vernachlässigen und für mehrschichtige Baumbestände, die zusätzlich veränderten Umweltbedingungen ausgesetzt sind, keine plausiblen Werte mehr zu liefern. Weiterentwickelte Modelle (Sloboda und Pfreundt (1989), Pfreundt (1988), Pfreundt und Sloboda (1996), Pretzsch (1992), Nagel et al. (2000), Guericke (2001)) versuchen über individuenbasierte Ansätze bessere Ergebnisse in Bezug auf Konkurrenzverhalten und Reaktion auf Umweltveränderungen zu erzielen. Das Individuum wird dabei sehr vereinfacht abgebildet, indem die Krone über einfache geometrische Körper wie z.B. achteckige unregelmäßige Pyramiden abgebildet wird. Diese Modelle sind alle in Software-Werkzeuge umgesetzt, die in prozeduralen oder objektorientierten Programmiersprachen implementiert sind. Die Modellparameter sind durchgehend empirisch ermittelt.

Andere, detailliertere Modelle (z.B. List et al. (1994), de Reffye et al. (1997), Perttunen et al. (1998), Eschenbach (2000)) versuchen, näher an Kausalitäten heranzukommen. Diese Modelle bilden die Architektur detailgenau ab und versuchen, über Licht- und Nährstoffallokationsmodelle das Wachstum zu beschreiben. Diese Ansätze sind in der Regel sehr rechenintensiv.

Die an forstlichen Fragestellungen orientierten Modellansätze von Pfreundt und Sloboda (1996), Pretzsch (1992), Nagel et al. (2000) und Guericke (2001) haben alle den Nachteil, sehr aufwändige Anforderungen an die Datenaufnahme zu stellen. Es stellte sich daher die Frage, ob es möglich ist, durch Aggregation eines der detaillierteren Einzelbaummodelle als Grundlage für die Bestandesmodellierung zu benutzen. Auf Grund der intensiven Kooperation mit der Finnischen Forstlichen Versuchsanstalt in Vantaa bei Helsinki ist das Modell LIGNUM (Perttunen et al., 1998) bei der Umsetzung zur Anwendung gekommen. Dabei wurde zusätzlich gezeigt, dass es möglich ist, derartige vereinfachte

Bestandesmodelle mit Lindenmayersystemen (Prusinkiewicz und Lindenmayer, 1990) zu implementieren. Diese haben vielerlei Vorteile hinsichtlich der Übersichtlichkeit und Nachvollziehbarkeit von Modellen.

Die Notwendigkeit, Modelle zu verifizieren bzw. zu falsifizieren, eröffnete die Frage nach einem Werkzeug in Form einer Software, das in der Lage ist, Modellergebnisse zu verarbeiten und zu analysieren. Die unterschiedlichen Formen der Ergebnisse, besonders, wenn dreidimensionale Architektur- und Bestandesstruktur-Informationen involviert sind, stellen besondere Anforderungen hinsichtlich der Repräsentation und adäquaten Auswertung der Modellergebnisse. Diese Anforderungen führten in unserem Projekt zu einer Umsetzung in ein wissenschaftliches Software-Werkzeug (GRODISC: GROgra-related DISCretisation tool) auf der Basis des objektorientierten Programmierparadigmas.

2 Lindenmayer-Systeme

Im Projekt sind zur Umsetzung von Modellen sogenannte Lindenmayersysteme (L-Systeme) benutzt worden (vgl. Prusinkiewicz und Lindenmayer (1990)). L-Systeme sind „String-rewriting-Systeme“. Das heißt, dass auf ein Startwort iterativ Regeln angewendet werden, die das Startwort verändern. In der Regel wird das Startwort dadurch pro Iterationsschritt länger. Der daraus resultierende String wird durch eine virtuelle „Turtle“ interpretiert. Die Turtle geht den String durch und sucht nach Kommandos, die sie ausführen kann, so denn welche spezifiziert sind. Das Ergebnis der Interpretation ist eine räumliche Struktur, die z.B. einen Baum abbilden kann.

Diese Interpretation wird auf dem Computer durch eine Software geleistet. In unserer Arbeitsgruppe wird dazu die Software GROGRA (Kurth, 1998) verwendet, die im Folgenden kurz beschrieben sei.

2.1 GROGRA

Zum besseren Verständnis seien hier einige Besonderheiten und die interne Datenhaltung beschrieben.

GROGRA ist ein Interpreter für erweiterte L-Systeme. Die Interpretation erfolgt in

zwei Schritten. Als erstes wendet GROGRA die spezifizierten Regeln auf ein Startwort an und wiederholt die Regeln so oft wie der Software befohlen ist. Der daraus resultierende String wird in einer zweiten Stufe von der Turtle in eine geometrische Struktur umgesetzt.

Die Regelspezifikation weist einige Besonderheiten auf. So ist es möglich, in die Regeln Zufallskomponenten einzubauen. So kann man etwa dem Austreiben eines Astes eine normalverteilte Austriebswahrscheinlichkeit zuordnen. Aber auch andere Zufallsverteilungen sind möglich.

Ein anderes sehr nützliches Hilfsmittel ist die Sensitivität. Durch diese wird eine zusätzliche Abhängigkeit in die Regelanwendung eingefügt. Nach der Erzeugung des Strings wird über eine zu spezifizierende Funktion die schon vorhandene geometrische Struktur aus dem vorherigen Schritt analysiert. Das ermöglicht, Bedingungen an das Wachstum einer Wachstumseinheit zu knüpfen, die z.B. den Abstand zu einer Nachbareinheit betreffen.

Die über die Turtleinterpretation entstandene geometrische Struktur wird in eine verkettete Liste von sogenannten *Sprosseinheiten* geschrieben. Jede dieser Einheiten verkörpert einen Schritt der Turtle. Die Einheit stellt geometrisch eine Strecke im dreidimensionalen Raum dar. Sie besitzt einen Anfangs- und einen Endpunkt. Zusätzlich werden in dieser Einheit Attribute gespeichert — im Wesentlichen die Länge, der Durchmesser und ein Blattparameter. Obwohl von GROGRA interpretierte L-Systeme ein sehr allgemeines Werkzeug sind, wird hier deutlich, dass dieses Werkzeug fast ausschließlich im Bereich der Pflanzenmodellierung eingesetzt wird.

GROGRA ist in der Programmiersprache C umgesetzt.

Für eine detaillierte Beschreibung von GROGRA und L-Systemen im Allgemeinen sei auf Kurth (1998) und Kurth (1994) verwiesen. Das letztere ist das Referenzhandbuch zu GROGRA und ist im Internet unter der folgenden Adresse verfügbar: <http://www.uni-forst.gwdg.de/~wkurth/grogra.html>.

2.2 Erweiterungen

Vorgesehen war die Simulation von Baumbeständen unter Zuhilfenahme von L-Systemen in der Sprachspezifikation des GROGRA-Interpreters (Kurth, 1998). Dies ist beispielhaft

an einem Kiefernbestand vollzogen. Die Baumbestände sollten aus einer Menge von Individuen bestehen. Die Einzelbäume sollten im Aufbau und im Konkurrenzverhalten vereinfacht dargestellt sein. Als Grundlage dafür diente das Einzelbaummodell von Pretzsch (1992).

Das Modell sieht einen vereinfachten Kronenaufbau in Form einer unregelmäßigen achteckigen Pyramide vor. Die Hauptachse ist ein Zylinder. Das Konkurrenzverhalten zwischen Individuen bestimmt sich aus dem Abstand einer unteren Ecke zur Kronenoberfläche eines Nachbarn. Die Richtung ist dabei durch den Pyramidenaufbau vorgegeben (vgl. Abbildung 2). Ein Überlappen der Kronen, das einem negativen Abstand entspricht, hat ein Zurückweichen der Krone zur Folge. Abbildung 1 und Abbildung 10 zeigen die Kronenstruktur in der Drauf- und Seitensicht.

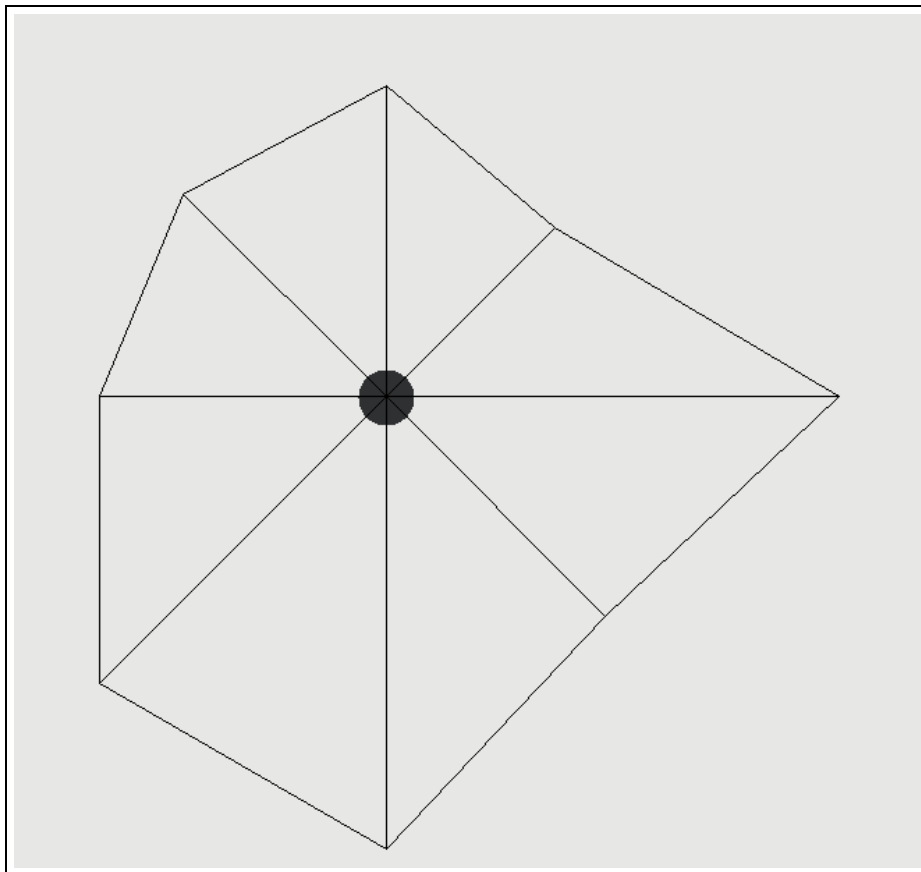


Abbildung 1: Die vereinfachte Krone in der Ansicht von oben.

Diese Vorgehensweise in der Modellierung bedingte die Notwendigkeit zu Änderungen

in der L-System-Spezifikation in zwei Punkten:

Als erstes ist es ineffektiv, eine unregelmäßige Pyramide ausschließlich durch Sprosseinheiten (Linien) abzubilden. Sinnvoller schien es, diese als System von Dreiecken darzustellen. Dazu wurde zur Turtlekommandosprache das Kommando $T(e_1, e_2, e_3)$ hinzugefügt. Dieses Kommando erzeugt ein Dreieck mit den Ecken e_1 , e_2 und e_3 . Diese werden als Index übergeben, der auf einen Punkt verweist. Dieser Index wird vorher über das Kommando $S(e_n)$ erzeugt.

Stellt man sich die Pyramide vor, so hat sie insgesamt neun Ecken: Der oberste Punkt, der die Baumspitze markiert, und die acht Ecken der Basisfläche. Von diesen Basisecken aus sollen die Abstände zu den Nachbarn berechnet werden. Diese sind aber durch die Dreiecke allein schlecht repräsentiert: Letztere stellen nur die Oberfläche dar. Die acht Basisecken entsprechen aber nicht nur schlicht einem Punkt, sondern auch einer Richtung. Es ist daher sinnvoll, diese durch eine Linie vom Stamm zur Ecke abzubilden. Wir nennen diese Linie Basiseinheit der Pyramide; sie wird als Sprosseinheit gespeichert. Aus Anfangs- und Endpunkt lässt sich die Richtung eindeutig bestimmen. Die Abbildung als Sprosseinheit eröffnet zusätzlich die Möglichkeit, Hilfsvariablen diesen Basiseinheiten zuzuordnen, um beispielsweise diese Einheit von der Stammeinheit in einem Baummodell zu unterscheiden.

Die zweite Änderung betrifft sensitive Funktionen. Die Berechnung des Abstandes zwischen einer Basiseinheit, die die untere Ecke einer unregelmäßigen Pyramide darstellt, und der Oberfläche eines Dreieckes war in den vorhandenen Funktionen nicht vorgesehen (vgl. Kurth (1998) und Kurth (1994)). Es ist daher eine neue Funktion (mit der Kenn-Nummer 14) zu der Sammlung hinzugefügt worden. Listing 1 zeigt die Funktionsdeklaration. Diese Funktion erhält beim Durchgehen durch die Liste der Sprosseinheiten die aktuelle Einheit übergeben. Sie berechnet dann für alle Dreiecke aus der globalen Dreiecksliste den minimalen Abstand zur aktuellen Einheit. Vorher ist bestimmt worden, ob es sich bei der Sprosseinheit um eine Basiseinheit einer unregelmäßigen Pyramide handelt.

Da Überlappung möglich ist, wurden negative Abstände zugelassen. Um zu vermeiden, dass Dreiecke berücksichtigt werden, die weit „hinter“ der Einheit liegen, werden nur Abstände berücksichtigt, die nicht kleiner sind als die negative Länge der Basiseinheit. Es

werden also, bildlich gesprochen, keine Überlappungen zugelassen, die über den Stamm hinaus gehen.

Die Funktion gibt den vorzeichenbehafteten Abstand zurück. Mit diesem Abstand ist es dann möglich, innerhalb des L-Systems zu arbeiten. Es ist etwa möglich, bei negativen Abständen einen Reduktionsfaktor für das Kronenwachstum umzusetzen.

Listing 1: Funktionsdeklaration der Funktion 14 (Berechnung des Abstands zwischen einer Sprosseinheit und der Oberfläche eines Dreiecks).

```
float function14(int ib, struct spross hu *current)
```

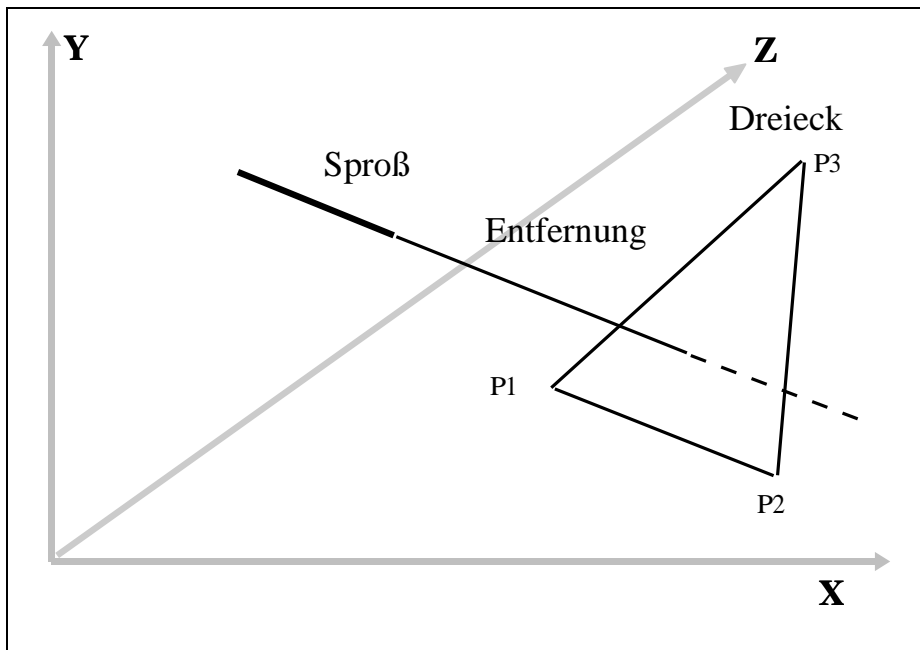


Abbildung 2: Abstand zwischen Sprosseinheit und Dreieck, das ein Teil einer Baumkrone repräsentiert.

3 GRODISC

3.1 Motivation und Aufgabe

Ziel des Projektes ist unter anderem die Erstellung eines Werkzeuges für die Analyse geometrischer Strukturen. Die geometrischen Strukturen sind Simulationsergebnisse von Modellen, die das Wachstum von Bäumen abbilden. Diese Strukturen sind in der Regel eine Menge von Elementareinheiten, die als Linien einen Baum oder einen Bestand von Bäumen abbilden. Es sind aber auch elementare Einheiten möglich, die nicht der geometrischen Figur einer Linie entsprechen, sondern z.B. die Krone vereinfacht über Dreiecke abbilden; vgl. Pretzsch (1992), Sloboda und Pfreundt (1989) und Pfreundt und Sloboda (1996).

Eine Analyse geometrischer Strukturen kann einzelne Zahlenwerte, Listen von Zahlen, aber auch selbst wieder eine komplexere Struktur liefern. Als Beispiel sei hier die Diskretisierung einer Wurzel in einem Voxelspace (kubisches Gitter) genannt (Oppelt et al., 2000). Ein Voxel (Volumen-Pixel, Würfel) wiederum ist ein geometrisches Objekt, das sich vollkommen von Linien und Dreiecken unterscheidet.

Ein Werkzeug in Form einer Software muss daher in der Lage sein, räumliche Daten ganz unterschiedlicher Natur zu halten. Neben der einfachen Haltung muss das Werkzeug diese räumlichen Strukturen mit reellwertigen Daten, wie die Länge eines Sprosses, verbinden. Es ergibt sich ein typisches Datenbankproblem. Es ist dabei dabei zu berücksichtigen, dass wir uns im dreidimensionalen Raum bewegen, wir aber auch (bei Zeitreihen) mit der vierten Dimension Zeit zu tun haben können.

Da das Werkzeug die Daten automatisiert analysieren soll, sind Optionen anzubieten, die eine Struktur möglichst allgemein analysieren. Diese Optionen sollten möglichst einfach erweiterbar sein.

Für die Lösung der Aufgabe teilt sich GRODISC in vier Teilbereiche:

- Die Datenstruktur,
- Algorithmen,
- Funktoren,

- Benutzerschnittstelle.

Die Umsetzung von GRODISC erfolgte objektorientiert unter Zuhilfenahme der Programmiersprache *C++* und der im Standard enthaltenen *Standard Template Library (STL)*. Für die Erklärung der Begriffe Algorithmen und Funktoren sei auf die entsprechenden Kapitel (s.u.) verwiesen. GRODISC ist vom Gesamtaufbau an LIGNUM (Perttunen et al., 1998) orientiert.

3.2 Die Datenstruktur

Die interne Datenstruktur von GRODISC muss hauptsächlich geometrische Strukturen abbilden und mit diskreten Daten verbinden. In der Welt der Software gibt es bereits Programme, die sich mit der Darstellung von geometrischen Strukturen und der Verbindung von Datenbanken mit diesen beschäftigen. Sie sind im allgemeinen als *Geographische Informationssysteme (GIS)* bekannt. Die Entwicklung von GRODISC hat sich daher an im GIS-Bereich übliche Datenstrukturen angelehnt, ist selbst aber kein vollständiges GIS. Die Software ist allerdings in der Lage, dreidimensionale Information zu benutzen. Die vierte Dimension Zeit ist ebenfalls in vereinfachter Form speicherbar.

Geometrische Strukturen werden in einem *GIS* einem abgegrenzten geographischen Raum zugeordnet und thematisch getrennt. Die thematische Trennung steht im direkten Zusammenhang zur Form des betrachteten geometrischen Körpers. So werden beispielsweise Flächen über Polygone und Städte durch Punkte im Raum dargestellt.

Die so abgegrenzten Themenkomplexe werden *Layer* (zu deutsch Schichten) genannt. Die Layer werden mit einer Datenbank verknüpft, die Attribute hält (z.B. die Einwohnerzahl von Städten).

GRODISC hat denselben Aufbau, um folgenden Ansprüchen an die Datenstruktur gerecht zu werden. Die Datenstruktur muss:

- die geometrische Struktur dreidimensional abbilden,
- die elementaren Einheiten sinnvoll miteinander verknüpfen,

- zusammengehörige Mengen von elementaren Einheiten eindeutig voneinander abgrenzen und
- dabei die Zeitkomponente zulassen.

Wir haben bereits festgestellt, dass GRODISC objektorientiert implementiert ist. Der Datenstruktur liegt somit eine Klassenhierarchie zugrunde. Abbildung 3 zeigt den Klassenaufbau.

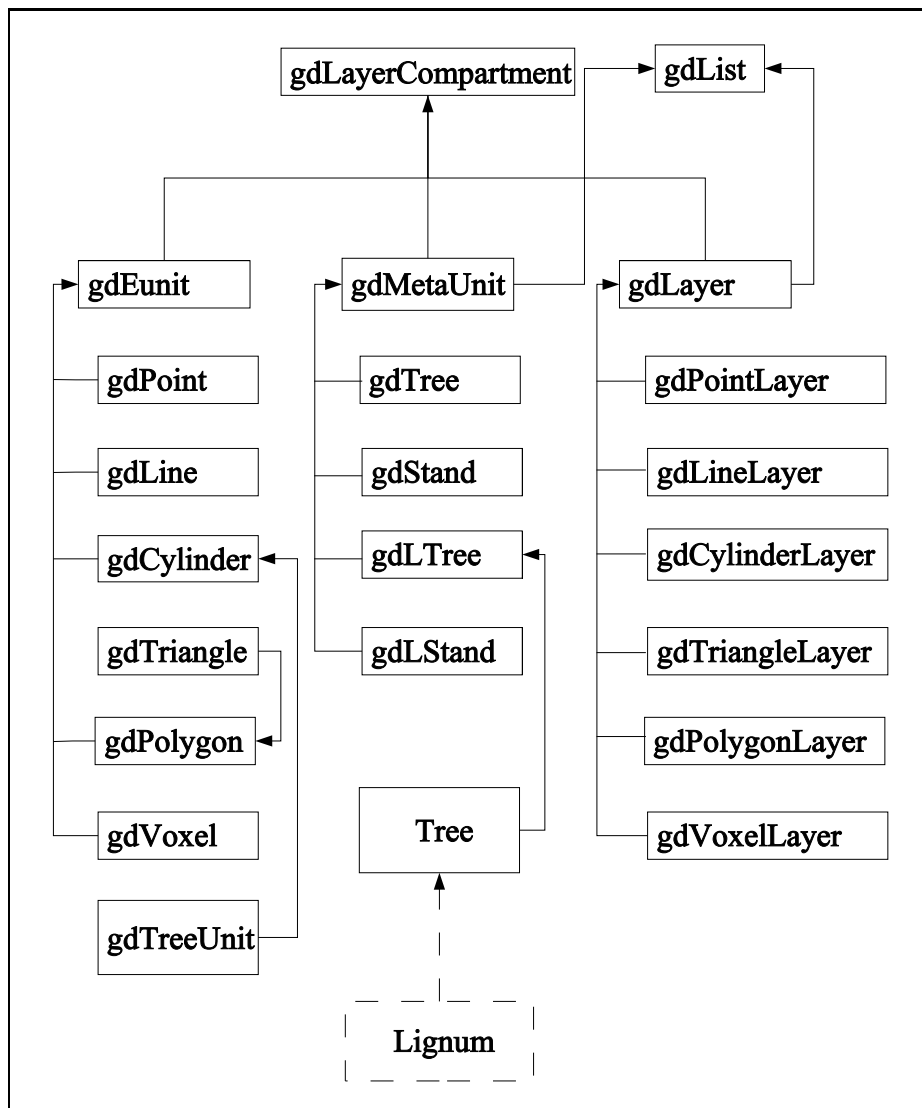


Abbildung 3: Die Klassenhierarchie in GRODISC

Grundlage für den Aufbau ist die Fokussierung der Software auf räumliche Einheiten.

Eine räumliche Einheit kann natürlich unterschiedliche Formen und Ausdehnungen haben. Es gibt aber Einheiten, die man nicht mehr unterteilen kann oder wo eine tiefere Unterteilung nicht mehr sinnvoll erscheint. Diese Einheiten werden hier als elementare Einheiten bezeichnet. Dies sind Punkte, Linien und Polygone. Modellergebnisse, die vereinfachte Kronenformen zum Ergebnis haben, machen es sinnvoll, zusätzlich zu allgemeinen Polygonen Dreiecke als elementare Einheiten anzusehen.

Der Übersicht und der besseren Abgrenzung wegen, werden in der höchsten Ebene, den *gdLayers*, nur einheitliche elementare Einheiten zusammengefasst. Das heißt, dass in den *gdLayers* nur räumliche Information eines bestimmten Types zusammengefasst ist. So gibt es pro Klassentyp einer elementaren Einheit nur eine Layerklasse. Die Layerklasse hält ihre Elemente in einer Liste.

GRODISC ist hauptsächlich als Anwendung in der Baum- bzw. Bestandesmodellierung gedacht. Dies macht es sinnvoll, zwei Klassen einzufügen, die in der Hierarchie genau zwischen den Layern und den Elementareinheiten liegen. Sie werden daher auch als *gd-MetaUnits* bezeichnet. Ein Baum z.B. ist zwar eine Menge von Linien. Es macht aber auch Sinn, eine Menge von Bäumen als Einheit anzusehen: Ein Bestand wäre eine solche Einheit. Die Klasse *gdTree* ist also eine Liste von Linien, gleichzeitig ist ein Baum aber auch Teil eines Layers (*gdTreeLayer*). Aus Gründen der Rechengeschwindigkeit und als Analogie zu GROGRA sind einige wichtige Attribute direkt den Linieneinheiten angefügt. Um die Klassenhierarchie nicht überzustrapazieren, ist hier mit der Klasse *gdTreeUnit* eine spezielle Baumeinheit entstanden. In ihr sind analog zur GROGRA-Sprosseinheit u.a. die Länge, der Durchmesser und ein Blattparameter gespeichert.

Viele Analyseoptionen haben einen Baumbestandesbezug. Diese erfordern einen besonderen Fokus auf Bestände auch in der Datenstruktur. Es ist daher ein Datentyp entstanden, der den Bestand repräsentiert. Die Klasse *gdStand* ist eine Liste von Bäumen. Gemäß des allgemeinen Aufbaus existiert ein Layer (*gdLStandLayer*), der eine Liste des Datentyps *gdStand* hält.

Es existieren im Aufbau zwei Klassen, die dem Import von LIGNUM-Klassen dienen. Das sind *gdLigTree* und *gdLigStand*. Sie sind mit den anderen beiden Metaeinheiten prinzipiell identisch. *L* steht hier für LIGNUM und bedeutet nichts anderes, als dass dieser

Baum von der Vererbung her ein LIGNUM-Baum ist (vgl. Abbildung 4). Der LIGNUM-Bestand fasst diese Bäume zusammen. Diese Klassen dienen der Datenschnittstelle zwischen GRODISC und LIGNUM. Ein GRODISC-Baum kann in einen LIGNUM-Baum überführt werden. Zusätzlich kann der LIGNUM-Baum durch seinen Aufbau erweiterte Analysemöglichkeiten eröffnen. Auch diese Klassen sind in einem Layer (*gdStandLayer*) zusammengefasst.

Jede Klasse erbt von *gdLayerCompartment*. Jede Klasse ist somit ein Kompartiment einer Schicht, inklusive der Schicht selbst. Jedes Kompartiment bekommt dadurch zusätzliche Funktionalität. Die wichtigste ist, dass Nachbarschaftsbeziehungen im Kompartiment als Attribute vorliegen. Die Klasse *gdLayerCompartment* enthält drei Listen, die Nachbarn als Elemente enthalten. Dies sind die Klassenelemente *_neighbours*, *_mothers* und *_daughters*. In *_neighbours* kann man alle Kompartimente ablegen, die man als Nachbarn ansieht — aber nur dann, wenn die Nachbarschaftsbeziehung symmetrisch ist. In realen Bäumen dagegen ist die Nachbarschaftsbeziehung zwischen den Austriebseinheiten gerichtet. In diesem Fall macht es Sinn, zwischen Nachbarn vor der Einheit und nach der Einheit zu unterscheiden. So hat eine Wachstumseinheit eine Mutter und eine bestimmte Anzahl von Töchtern. Die Mutter wäre einziges Mitglied der Liste *_mothers*, die Töchter Elemente von *_daughters*.

Zum Schluss sei noch ein technischer Hinweis gegeben. Es macht keinen Sinn, die vielen Listen als Listen der instanziierten Objekte selbst zu halten. Sie würden sehr groß und unhandlich. Das trifft insbesondere auf die Nachschäftslisten des *gdLayerCompartment* zu. Die Klasse *gdList*, die die Haltung übernimmt, ist daher eine *STL-Liste* von Zeigern auf die Objekte.

3.3 Algorithmen

Ziel der Erstellung eines Werkzeuges ist die Entwicklung von Analysealgorithmen für die geometrische Struktur. Diese sollen möglichst übersichtlich, elegant und einfach erweiterbar sein.

Die Datenstruktur von GRODISC ist listenorientiert. Um Analysen der Struktur durchführen zu können, muss man jedes Listenelement auf eine sinnvolle Art und Weise

ansteuern und auf das Element eine Vorschrift anwenden können. Die Umsetzung der Algorithmen ist in diese zwei Teilbereiche aufgeteilt: Das eigentliche „Ansteuern“ des Elementes und die Ausführung einer Vorschrift an diesem Element. Das „Anfahren“ von Listenelementen wird hier über Funktionen gesteuert. Diese beruhen auf dem Design der *STL*; die *STL* bietet solche Funktionen an. Als Beispiel für eine derartige Funktion sei hier die *for_each*-Funktion genannt. Abbildung 2 zeigt den Aufbau. Der Funktion werden zwei Iteratoren übergeben, die den Anfang und das Ende der Liste repräsentieren. Zusätzlich wird der Funktion eine Vorschrift in Form eines Zeigers auf eine Funktion übergeben. *for_each* wendet diese Vorschrift dann auf das Listenelement an. Die Kombination aus Durchlaufen der Liste und Anwendung einer Vorschrift ergibt den Algorithmus.

Listing 2: Der *for_each* Algorithmus der *STL*

```
template <class It , class Fun>
void for_each(It first , It last , Fun fun){
    while( first != last ) fun (*first ++);}

```

Andere Algorithmen der *STL* liefern zusätzlich einen Wert zurück. So kann man z.B. Werte einer Liste aufaddieren. Der dazu benötigte Algorithmus heißt *accumulate*. (Vergleiche hierzu Stroustrup (1998).)

GRODISC selbst bietet ebenfalls Algorithmen dieser Art an. Sie sind zwar an die *STL*-Algorithmen angelehnt bzw. benutzen sie sogar, sind aber naturgemäß komplizierter. Das liegt an der Baumstruktur, die parallele Listen enthält, die man nicht einfach sequenziell durchlaufen kann. Den Algorithmen in GRODISC werden zusätzlich andere Parameter übergeben. In der *STL* werden der erste und der letzte Iterator der Liste übergeben. Hier wird gleich die Referenz zum untersuchenden Objekt dem Algorithmus präsentiert. Als zusätzlichen Parameter kann man angeben, ob man Ebenen oder Objektklassen definieren will, die man auslassen möchte. In GRODISC existieren folgende Algorithmen:

- ForEach,
- Accumulate,

- Compare,
- Find,
- FindIf,
- Found.

Übergabeparameter im Einzelnen sind:

- ein Zeiger auf *gdLayerCompartment*,
- ein Parameter zur Bestimmung der auszulassenden Elementklassen.

Zusätzlich muss dem Algorithmus *Accumulate* als zweiter Übergabeparameter derjenige Wert übergeben werden, der verändert werden soll. *Accumulate* dient als Abfragefunktion. Die übergebene Funktion regelt, welche Werte abgefragt werden sollen und welche Werte zurückgegeben werden.

Neu ist hier der *Compare*-Algorithmus. Bei vielen Berechnungen kam es darauf an, vergleichende Funktoren zu entwickeln. Dazu muss aus einer Menge von Einheiten eine Einheit entnommen werden. Diese wird dann mit allen Elementen einer anderen Menge von Einheiten verglichen, wobei die Vergleichsoperatoren spezifiziert werden müssen. Es ist möglich, dass beide Mengen identisch sind. Als Beispiel sei hier die Berechnung der n nächsten Nachbarn eines Baumes genannt. Dazu muss man einen Baum aus einem Bestand betrachten und alle Abstände zu den noch vorhandenen Bäumen im Bestand berechnen.

Find und FindIf sind analog zur *STL*. Diese Funktionen suchen nach Objekten innerhalb der Listen. FindIf kann man zusätzlich einen Vergleichsoperator übergeben. Found funktioniert genauso, liefert aber nur zurück, ob ein Objekt gefunden wurde oder nicht. Diese Funktion bricht den Suchalgorithmus ab, wenn das Objekt gefunden ist.

3.4 Funktoren

Den Algorithmen werden Vorschriften in Form von Zeigern auf Funktionen übergeben. Die Deklaration des Funktionsübergabeparameters als Template erlaubt es, die Anwen-

dungsvorschrift auch anders zu gestalten.

Nachteil dieser Vorgehensweise ist nämlich, dass die Anzahl der Übergabeparameter begrenzt ist. In GRODISC sind deshalb viele dieser Vorschriften als Klammeroperator innerhalb einer Struktur umgesetzt. Dadurch ist man in der Lage, über statische Variablen der Struktur zusätzliche Steuerelemente in einer Funktion zu integrieren, ohne der Funktion die Parameter direkt zu übergeben. Derartige Funktionen werden auch als *Funktoren* bezeichnet (Stroustrup, 1998).

An dieser Stelle sei ein Beispiel für einen derartigen Funktor mit statischen Variablen gegeben. Der Funktor *CalcNextNeighbours* berechnet die nächsten Nachbarn eines *gdLayerCompartments*. Diesen Funktor kann man z.B. dazu benutzen, die drei nächsten Nachbarn eines Baumes zu ermitteln. Das ist ein typischer Funktor für den Compare-Algorithmus. Der Funktor bekommt in unserem Beispiel zwei Bäume übergeben, wobei der erste den Baum darstellt, für den man die nächsten Nachbarn berechnen möchte, während der zweite derjenige ist, für den man bestimmen möchte, ob er zu der Gruppe der nächsten Nachbarn dazugehört.

Das Problem stellt sich hier, dass die Anzahl der nächsten Nachbarn variabel sein sollte. Man hat aber für den Funktor nur zwei Übergabeparameter. Es ist also unmöglich, die Anzahl der nächsten Nachbarn mit zu übergeben. Zur Lösung dieses Problems ist *CalcNextNeighbours* als Struktur umgesetzt, die die statische Variable *num_neighbours* enthält, die standardmäßig auf eins gesetzt ist. Man kann vor dem Start des Compare-Algorithmus diesen Wert verändern. So ist es z.B. möglich, die Dreipunkt-Methode von Földner (1995) umzusetzen, bei der man die drei nächsten Nachbarn eines Baumes benötigt, in dem man *CalcNextNeighbours::num_neighbours* den Wert drei zuordnet.

3.5 Datenschnittstellen

Für den Teilbereich der Modellaggregation, aber auch für den Modellvergleich war es wichtig, den Datenaustausch zwischen GRODISC und anderen Modellierungstools herzustellen. Dazu wurden Schnittstellen zu LIGNUM (Perttunen et al., 1996) und AMAPmod (Godin et al., 1998) implementiert. Zu diesem Themenkomplex sind zwei Veröffentlichungen entstanden (Dzierzon und Kurth (2001a) und Dzierzon et al. (2001b)). Die Schnitt-

stellen seien hier nur kurz umschrieben. Die Artikel sind diesem Bericht beigelegt.

Für die Schnittstelle nach LIGNUM galt es zwei Problembereiche zu lösen: Die Einbindung der LIGNUM-Datenstruktur und der eigentliche Übersetzungsprozess. Der Aufbau von LIGNUM hat Ähnlichkeiten zu GRODISC — das ist kein Zufall: LIGNUM ist Vorbild für den Aufbau von GRODISC gewesen. Die Abbildung 4 zeigt die Klassenhierarchie.

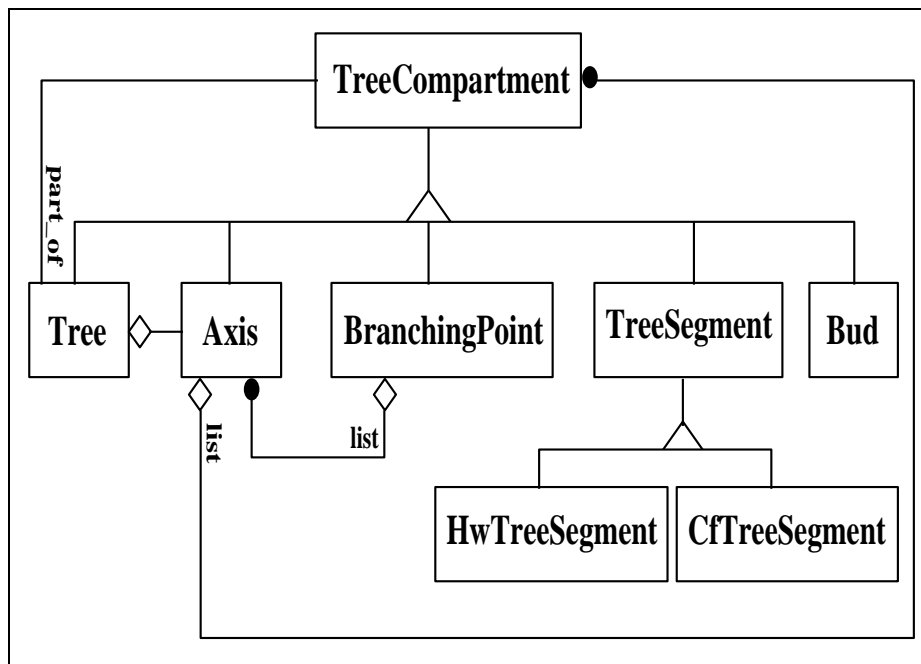


Abbildung 4: Die Klassenhierarchie von LIGNUM (Perttunen et al., 1998).

Auch hier gibt es ein zentrales Element in Form des *TreeCompartments*. Auch hier ist der Baum durch eine *STL-Liste* umgesetzt. Und auch hier gibt es an die *STL* angelehnte Algorithmen. Die Übersetzung von dieser Struktur in eine GRODISC-Struktur und umgekehrt erfolgt über einen sogenannten *PropagateUp*-Algorithmus. Dieser gibt einem die Möglichkeit, die Gravelius-Ordnung zu berechnen, die nicht notwendig in der LIGNUM-Struktur abgespeichert ist. Das Ergebnis der Übersetzung ist entweder eine GRODISC-Struktur oder eine GROGRA und GRODISC verständliche ASCII Datei im sogenannten DTG-Format.

Die Schnittstelle zu AMAPmod (Godin et al., 1998) soll dazu dienen, die Analysemöglichkeiten für eine erzeugte dreidimensionale Struktur zu erweitern (Dzierzon und Kurth, 2001a). AMAPmod ist eine Software, die eine Abfragesprache enthält, die ähnlich einer

Datenbankabfrage funktioniert. Es hat sich einfach angeboten, diese Ressource zusätzlich zu nutzen.

3.6 Benutzerschnittstelle

Für den Gebrauch ist es notwendig, eine Schnittstelle zum Benutzer zu erstellen. Zu Beginn der Überlegungen stand die Frage, wie eine derartige Schnittstelle umzusetzen ist. Das Problem ist allseits ähnlich. Das Programm soll mindestens von allen Menschen in der Arbeitsgruppe genutzt werden können, ohne sich viel in die Tiefen des Programmsystems einarbeiten zu müssen. Dieses erreicht man am besten über eine Grafische Benutzeroberfläche *GUI*. Eine Vorgabe für die Software war, die Plattformunabhängigkeit zu gewährleisten. Dies gestaltet sich aber im Bereich des *GUI* sehr schwierig, weil ein *GUI* Fenstersysteme benutzt, die von Betriebssystem zu Betriebssystem sehr unterschiedlich auch in der Programmiersyntax sind. Wir haben zur Lösung des Problems auf Klassen der *qt*-Bibliotheken zurückgegriffen. *qt* stammt von Trolltech in Norwegen. Diese Klassen sind ein ausgezeichnetes Tool zur Umsetzung von plattformunabhängigen *GUI*. Sie unterliegen für X11-Window-Systeme der OpenSource-Lizenz und sind somit für den nichtkommerziellen Gebrauch frei verfügbar.

Die Abbildung 5 zeigt einen Screenshot der Benutzeroberfläche von GRODISC. GRODISC in z.Zt. unter SuSE Linux 7.0 und 7.2 implementiert. Die Fensterklassen sind mit Hilfe der qt3.0.0 Distribution von Trolltech AS aus Norwegen erstellt. Die Abbildung der Objekte erfolgt unter Zuhilfenahme der OpenGL®-Bibliotheken.

3.7 Diskussion

Es hat sich gezeigt, dass die Datenstruktur insbesondere in Zusammenhang mit dem Design der Analysealgorithmen ein gutes Werkzeug ist, dreidimensionale Strukturen zu analysieren. Das Design der Analysealgorithmen ermöglicht insbesondere ein schnelles und elegantes Erweitern und Hinzufügen von Analyseoptionen. Der Erstellung von Funktoren hat sich dabei als gewöhnungsbedürftig, letztlich aber als übersichtlich herausgestellt. So braucht man nicht für jedes Objekt neue Funktionen zur Visualisierung zu erstellen,

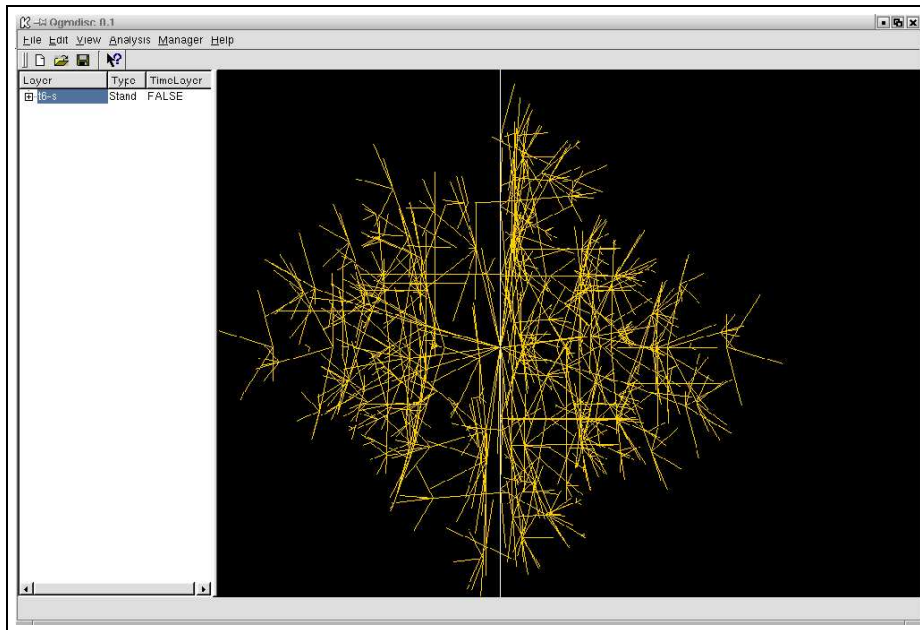


Abbildung 5: Screenshot der graphischen Benutzerschnittstelle von GRODISC.

sondern kann dies innerhalb eines Funktors abwickeln.

Die Datenstruktur als solche weist noch einige Unzulänglichkeiten auf. Im Prinzip ist es ein sinnvoller und übersichtlicher Weg, 3D-Geometrien in einer Klassenhierarchie abzubilden. Handlungsbedarf besteht in der Speicherverwaltung. Manche Strukturen sind zu umfangreich, um komplett im Arbeitsspeicher gehalten zu werden. GRODISC hat bis jetzt keine Möglichkeit, Daten innerhalb einer Datei zu verwalten. Es ist daher angedacht, große Datenmengen, die nicht in den Arbeitsspeicher passen, über ein kleines Datenbanksystem zu verwalten.

Die Benutzeroberfläche hat sich ebenfalls als speicheraufwändig herausgestellt. Es scheint das System schnell zu überlasten, wenn ein Baum mit über 20000 Einheiten grafisch mit *OpenGL* visualisiert werden soll. Eine Möglichkeit, dies etwas effektiver zu gestalten, wäre die Programmierung eines Textparsers ähnlich AMAPmod. Das eröffnet die Möglichkeit, eine Struktur in den Arbeitsspeicher zu heben, ohne gleich mit aufwändigen Grafiken und Benutzeroberflächen hantieren zu müssen. Ein Parser würde auch die Flexibilität bei der Handhabung von Strukturen erhöhen.

4 Rohdatenerhebung

4.1 Motivation

Ziel des Projektes war es unter anderem, ein Werkzeug zu schaffen, das dem Modellvergleich, aber auch zur Bewertung der Modellgüte dient. Das Werkzeug sollte beispielhaft an vorhandenen Modellen bzw. aggregierten Modellen getestet werden. Da ein Modell die Realität abbildet, muss es Ziel sein, zumindest einen Ausschnitt aus der Realität zu kennen, um einen Vergleich zwischen Modellergebnis und Realdaten durchführen zu können.

Ein Modell, das hier zur Anwendung kam, war eine Kiefernversion (*Pinus sylvestris*) von LIGNUM (Perttunen et al., 1998).

Es war nicht vorgesehen, selbst Rohdatenmaterial zu erheben. Es sollte auf schon vorhandenes Datenmaterial zurückgegriffen werden. Wir brauchten für den Vergleich folgende Daten:

- BHD,
- Höhe,
- Kronenansatz,
- Kronenablotungen in acht Richtungen,
- Positionsdaten.

Für die Untersuchung der Entwicklung der Parameter war es außerordentlich wichtig, wenigstens eine Wiederholungsmessung zu haben.

Es hat sich bei der Recherche nach schon vorhandenen Daten gezeigt, dass schon gemessene Bestandesdaten dem Anspruchsprofil nie entsprachen. Wir hatten dazu Kontakt mit mehreren Institutionen: Der Finnischen Forstlichen Versuchsanstalt, der Niedersächsischen Forstlichen Versuchsanstalt, der Universität Helsinki, dem Institut für Forsteinrichtung und Ertragskunde an der Universität Göttingen und dem Lehrstuhl für Waldwachstumskunde an der TU München. Es sind zwar Messungen an Kiefer durchgeführt

worden, sie entsprachen aber in keinem Fall dem benötigten Profil. Es fehlten sowohl Wiederholungsmessungen als auch Nachbarschaftsrelationen.

Einzige Ausnahme bildete ein Datensatz, der 1996 von Frau M.-S. Duchiron (Universität Nancy) in Norddeutschland erhoben wurde und eine Datengrundlage ihrer Dissertation (Duchiron, 2000) darstellte. Diese Daten enthielten alle Anforderungen bis auf die Wiederholung. Da zwischen der ersten Erhebung und dem Zeitpunkt der Recherche sechs Jahre lagen, haben wir uns dazu entschlossen, eine erneute Vermessung der Daten vorzunehmen. Dieses ist dann im Oktober 2001 geschehen.

4.1.1 Die Untersuchungsfläche

Die Untersuchungsfläche lag in der Norddeutschen Tiefebene bei Syke. Syke liegt südlich von Bremen. Der Untergrund ist ein armer Sandboden, auf dem ein reiner Kiefernbestand stockt (vgl. Duchiron (2000)). Dieser ist 1950 mit klassischen Pflanzverfahren begründet. Er ist somit 1996 42 und 2001 47 Jahre alt gewesen. Näher beschreibende Parameter des Bestandes wie Durchmesser und Zuwächse sind im Ergebnisteil wiedergegeben.

4.1.2 Aufnahmeparameter und Messmethoden

Tabelle 1 gibt die Aufnahmeparameter wieder. Es waren im Einzelnen Durchmesser als BHD, die Höhe des Baumes, der Kronenansatz und acht Ablotungen der Kronenausdehnung.

Parameter	Einheit	Präzision
BHD	cm	mm
Höhe	m	cm
Kronenansatz	m	cm
Kr.-Ausdehnung	m	cm

Tabelle 1: Aufnahmeparameter

Zur Anwendung kamen der Suunto-Neigungsmesser für die Bestimmung der acht Richtungen für die Kronenablotung, ein Umfangsmessband für die Messung der Durchmesser

und ein Kronenspiegel für die Ablotung der Kronendurchmesser. Für die Höhenmessung ist ein Blume-Leis-Höhenmesser verwendet worden. Es sei an dieser Stelle der Niedersächsischen Forstlichen Versuchsanstalt gedankt, die die Geräte freundlicherweise zur Verfügung gestellt hat.

Die X- und Y-Koordinaten sind 1996 von Marie-Stella Duchiron (Duchiron, 2000) eingemessen worden. Für die räumliche Verteilung hat sie ein sog. Top-Con benutzt, mit dem man in Zentimetergenauigkeit die Baummittelpunkte im Verhältnis zu einem definierten Ursprung vermessen kann.

4.2 Ergebnisse

Im Folgenden sollen die Messergebnisse dargestellt werden. Die Abstände zu den nächsten Nachbarn sind nachträglich von GRODISC berechnet worden.

4.2.1 Allgemeine Statistiken

Die Tabellen 2, 3 und 4 geben eine Übersicht über die Messergebnisse. Die Werte für die Ablotungen und die Distanzen zu den nächsten Nachbarn sind hier der Übersicht halber aggregiert als Mittelwert über die acht Himmelsrichtungen dargestellt.

Variable	N	Mean	Std Dev	Minimum	Maximum
BHD	151	18.6	4.2	10.3	31.3
Höhe	151	18.0	2.0	11.5	24.0
KA	151	12.3	1.5	0.0	15.0
KRMEAN	151	1.5	0.6	0.0	2.9
DMEAN	80	1.7	3.2	-4.4	13.5

Tabelle 2: Teilbestandeskennwerte 1996. KA = Kronenansatz, KRMEAN = Mittlerer Kronenradius pro Baum, DMEAN = Mittlerer vorzeichenbehafteter Abstand zum nächsten Nachbarn pro Baum

Die Werte bedürfen der Kommentierung. Als erstes sei hier die Messung der Höhen genannt. Die Zuwächse sind im Mittel zwar positiv, es existieren aber vereinzelt negative

Variable	N	Mean	Std Dev	Minimum	Maximum
BHD	94	23.3	4.3	14.8	36.0
Höhe	94	20.0	1.8	16.0	24.5
KA	94	13.8	1.3	10.5	17.5
KRMEAN	94	2.2	0.6	1.2	4.7
DMEAN	51	1.1	3.7	-3.9	14.6

Tabelle 3: Teilbestandeskennwerte 2001. KA = Kronenansatz, KRMEAN = Mittlerer Kronenradius pro Baum, DMEAN = Mittlerer vorzeichenbehafteter Abstand zum nächsten Nachbarn pro Baum

Variable	N	Mean	Std Dev	Minimum	Maximum
BHD	94	2.9	1.1	0.6	7.0
Höhe	94	1.7	1.7	-2.5	6.0
KA	94	1.5	1.3	0.0	6.5
KRMEAN	94	0.5	0.6	1.2	4.7
DMEAN	93	2.9	4.9	-3.6	24.0

Tabelle 4: Teilbestandeskennwerte für die Zuwächse. KA = Kronenansatz, KRMEAN = Mittlerer Kronenradius pro Baum, DMEAN = Mittlerer vorzeichenbehafteter Abstand zum nächsten Nachbarn pro Baum

Zuwächse. Daher der Wert von -2.5 als Minimum an Höhenzuwachs. Recherchen haben den Grund nicht völlig klären können. Es scheint so zu sein, dass einige Bäume durch Fällungsarbeiten in der Krone geschädigt waren, was natürlich eine Reduktion der Höhe zu Folge haben kann. Bei anderen Bäumen scheint aber ein Messfehler vorzuliegen. Dafür gibt es zwei Erklärungsansätze:

Die Individuen auf der Fläche sind 1996 zur Identifikation mit Pappnummern am Stamm versehen worden. Diese waren 2001 komplett (!) verschwunden. Die Positionen der Einzelbäume mussten daher innerhalb des Bestandes rekonstruiert werden. Wir sind uns sehr sicher, dass wir die Bäume wiedergefunden haben. Letztliche Sicherheit besteht

aber natürlich nicht.

Eine weitere Fehlerquelle ist die Handhabung des Blume-Leis-Höhenmessers. Die Bäume sind von zwei verschiedenen Personen vermessen worden, die zusätzlich das Individuum mit hoher Wahrscheinlichkeit von verschiedenen Richtungen angepeilt haben. Das kann unterschiedliche Einschätzungen der Lage des Stammfußes und des obersten Punktes der Stammkrone zu Folge haben. Stellt man sich einen Baum innerhalb eines dichten Bestandes vor, so wird klar, dass die Baumkrone nicht aus allen Richtungen gleichmäßig sichtbar ist. Bei der Anpeilung der obersten Spitze kann es durchaus sein, dass man von einer Seite aus einen Ast übersieht oder fälschlicherweise einem anderen Baum zuordnet, während man aus einer anderen Richtung diese durchaus richtig eingeschätzt hätte. Ähnliches gilt für den Stammfuß. Er kann aus verschiedenen Richtungen durchaus unterschiedlich im Bodenbewuchs verschwinden, was ebenfalls Messfehler bedingen kann. Da der Bestand relativ dicht und der Bodenbewuchs sehr stark war, kann man davon ausgehen, dass derartige Messfehler entstanden sind.

Des Weiteren ist die Standardabweichung der Abstände zum nächsten Nachbarn sehr hoch, insbesondere bei den Zuwächsen. Grundlage für die Bestimmung der Abstände sind die Kronenablotungen. Diese als solche sind problematisch in Bezug auf die Präzision der Messung. Die hohen Maximalwerte erklären sich aus Randeffekten. Ein Baum, der am Rand des Teilkollektivs steht, hat in bestimmte Richtungen entweder keinen Nachbarn oder es kann sein, dass ein Nachbar nach der Definition auch am Rand, aber sehr weit weg steht.

Zwischen 1996 und 2001 hat eine Durchforstung stattgefunden, bei der 57 Bäume geerntet wurden. Das entspricht etwa einem Drittel.

4.2.2 Korrelationsanalysen

Für die Umsetzung eines Wachstumsmodelles wurden die Bestandesparameter untereinander in Beziehung gesetzt. Es galt die Hypothese zu bestätigen oder zu widerlegen, dass das Wachstum aller gemessenen Individualparameter in Beziehung zum Durchmesser des Einzelbaumes steht. Oder, anders gesagt, wir haben eine Korrelation zwischen den Durchmessern der Bäume und den Höhen, Kronenansätzen und durchschnittlichem

Kronendurchmesser erwartet — und zwar sowohl für 1996 und 2001 als auch für die Zuwächse.

Es sollen in diesem Bericht die Ergebnisse dieser Analysen beispielhaft am Zusammenhang zwischen Durchmesser und durchschnittlichem Kronendurchmesser dargestellt werden. Für die anderen Korrelationen ergaben sich ähnliche Ergebnisse.

Die Abbildungen 6 und 7 zeigen die Güte der Anpassung für die Jahre 1996 und 2001. Man erkennt einen deutlichen Zusammenhang mit einer Irrtumsscheinlichkeit von $p < 0.0001$. Anders bei den Zuwächsen (Abbildung 8). Die Abbildung sagt eigentlich schon alles. Hier ist kein korrelativer Zusammenhang erkennbar ($R^2 = 0.00$).

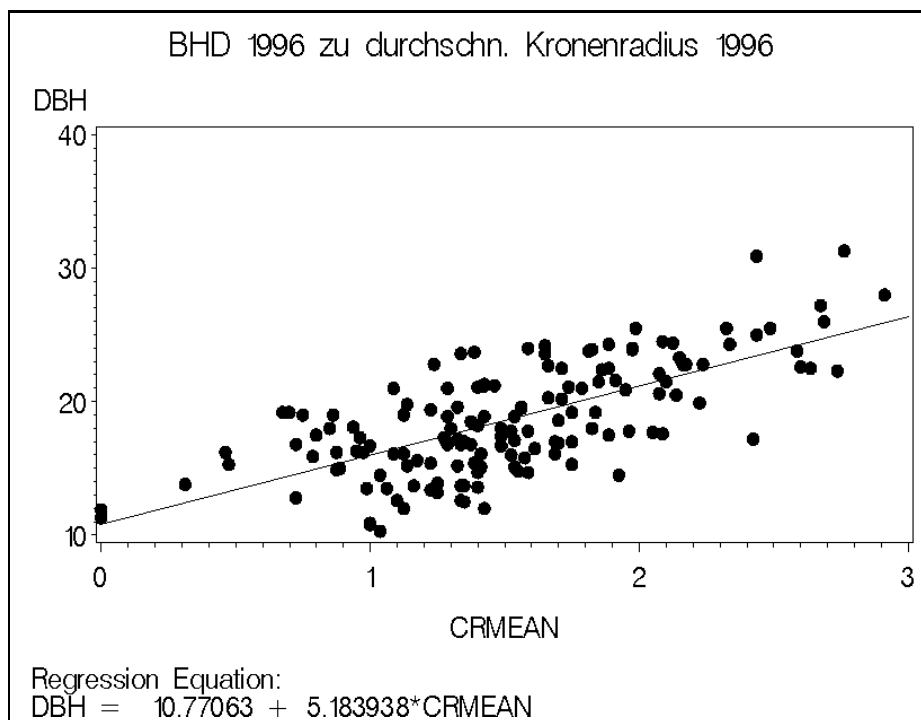


Abbildung 6: Der Zusammenhang zwischen Durchmesser und durchschnittlichem Kronendurchmesser 1996 ($R^2 = 0.49$, $p < 0.0001$). DBH = BHD, CRMEAN = durchschnittlicher Kronendurchmesser.

4.2.3 Analyse der Kronenabstände

Das Modell, das auf Realdaten beruht, hat eine Konkurrenzkomponente: Es wurde eine Abstandssensitivität eingebaut. Schon Pretzsch (1992) hatte eine derartige Sensitivität

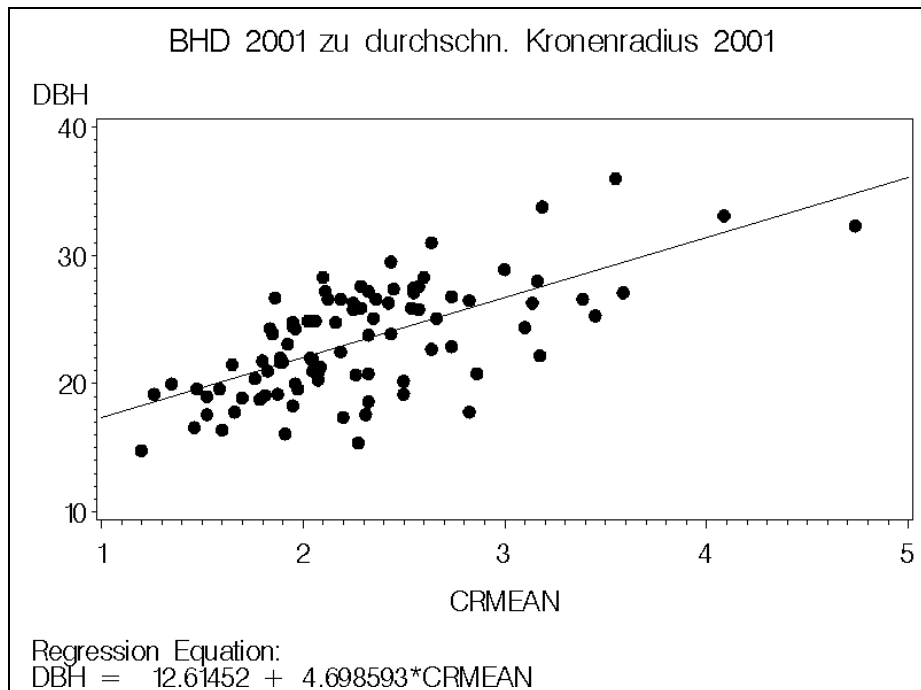


Abbildung 7: Der Zusammenhang zwischen Durchmesser und durchschnittlichem Kronendurchmesser 2001 ($R^2 = 0.44$, $p < 0.0001$). DBH = BHD, CRMEAN = durchschnittlicher Kronendurchmesser.

postuliert. Die Methode von Pretzsch konnte hier aber nicht völlig umgesetzt werden, da sich nicht eindeutig gezeigt hat, ob ein Zurückweichen einer Krone wirklich in einem korrelativen Zusammenhang zum Abstand zum Nachbarn steht. Dies ließ sich hier nicht nachweisen. Wir konnten aber zeigen, wie Abbildung 9 verdeutlicht, dass zumindest ein Minimalabstand besteht. Das Kronenwachstum scheint also in diesem Bestand ab einem bestimmten Abstand abzurechnen.

5 Ein Bestandesmodell für Kiefer

5.1 Der Einzelbaum

Das Einzelbaummodell beruht auf dem Design von Pretzsch (1992) für die Modellierung von Fichten (*Picea abies*) im Bestand. Pretzsch (1992) hat den Baum in einer sehr vereinfachten Art und Weise abgebildet: Der Stamm ist schlicht ein Zylinder, und die Krone

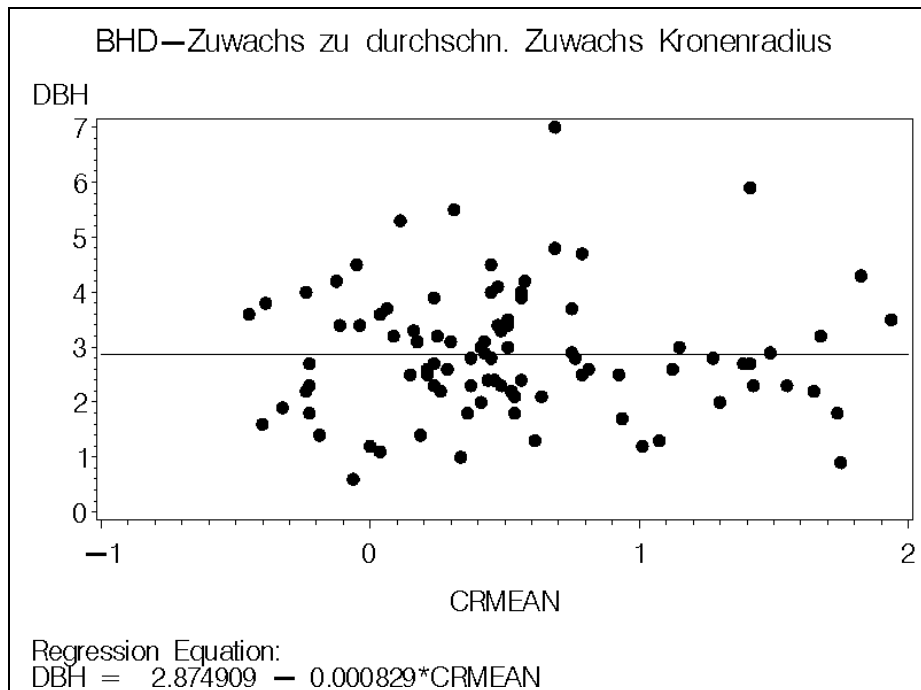


Abbildung 8: Der Zusammenhang zwischen Durchmesser und durchschnittlichem Kronendurchmesser bei den Zuwächsen ($R^2 = 0.00$, $p = 0.9968$). DBH = BHD, CRMEAN = durchschnittlicher Kronendurchmesser.

wird als eine unregelmäßige Pyramide mit acht Ecken repräsentiert. Die Ecken sind in die Himmelsrichtungen ausgerichtet. Es gibt eine Ecke, die genau nach Norden zeigt, die anderen richten sich jeweils im 45° Winkel aus (vgl. Abbildung 1 und Abbildung 10).

Der Bestand soll über die Zeit modelliert werden. Das Wachstum ist abhängig von der Interaktion zwischen Individuen. Die Darstellung des Modells in Bezug auf das Wachstum soll daher nach der Beschreibung der Interaktion erfolgen.

5.2 Interaktion

Es soll in diesem Modell ein kompletter Bestand abgebildet werden. In einem Bestand findet Konkurrenz statt: Das bedeutet, es gibt Interaktionen zwischen den Baumindividuen. Diese Interaktionen sind ebenfalls einfach abgebildet. Pretzsch (1992) postuliert eine Abhängigkeit zwischen der Ausdehnungsrate der Kronen und dem zu einem Zeitpunkt existierenden Abstand zu einer Nachbarkrone – negativer Abstand bedeutet Überlappung.

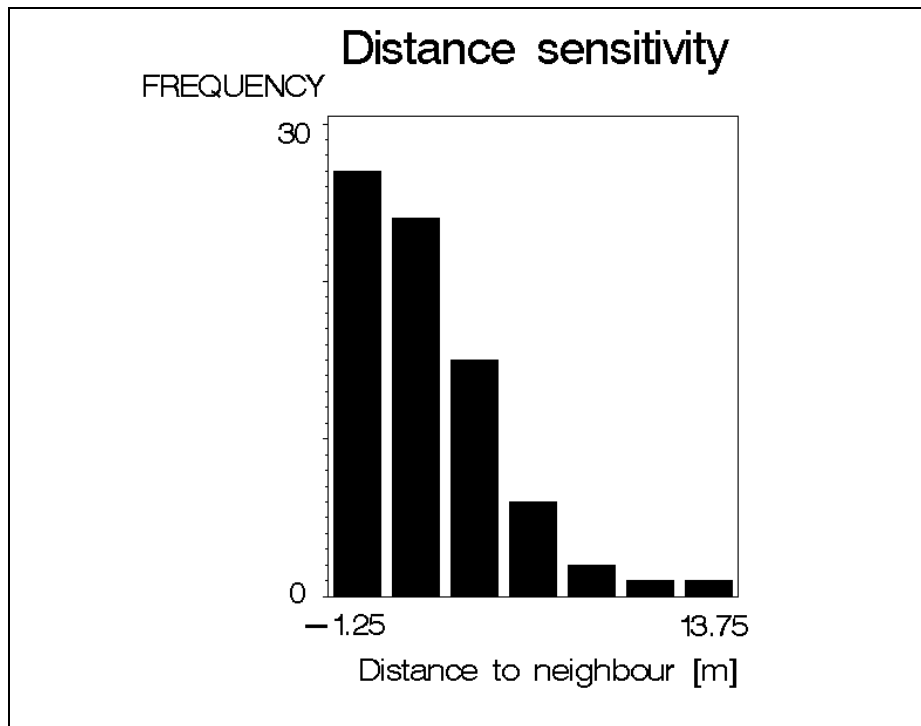


Abbildung 9: Die durchschnittlichen Kronenabstände.

Der Abstand wird als der kleinste Abstand zu einem Nachbar in die Richtung der jeweiligen Ecken der unregelmäßigen Pyramide berechnet. Die Kronenausdehnungsraten werden also pro Ecke berechnet. Tritt eine Überlappung auf, so reagiert die Krone mit einem Rückgang der Kronenausdehnung. Dieses Verhalten ist anhand von Realdaten statistisch angepasst worden.

Der Versuch, dieses Modell vollkommen auf unsere Kieferndaten anzupassen, ist gescheitert. Die Kiefer zeigte keinerlei Verhalten, in das sich ein Zurückziehen der Krone bei Überlappung hineinleiten ließe. Die Abbildung 9 hat aber gezeigt, dass trotzdem eine gewisse Berührungssensitivität besteht. Dies bestätigt Beobachtungen aus dem Bestand selbst, bei denen man den deutlichen Eindruck hatte, dass Kiefern keinen „Hang“ zur Überlappung haben. Im Modell ist das dadurch abgebildet, dass die Kiefer ab einem bestimmten Überlappungsgrad keine Kronenausdehnung mehr zeigt.

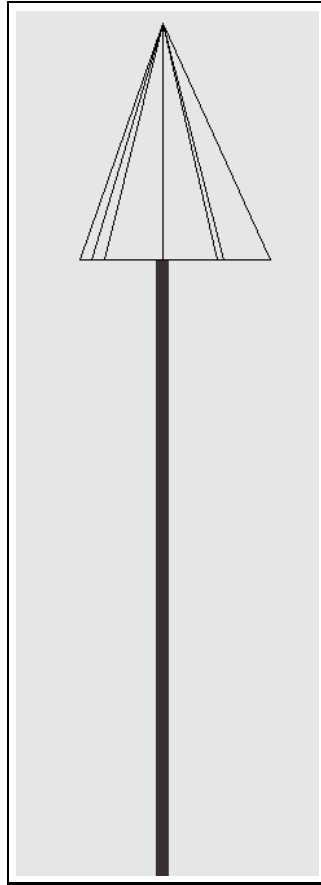


Abbildung 10: Die vereinfachte Krone in der Seitensicht.

5.3 Das Wachstum

Die das Wachstum betreffende Hypothese war, dass man Abhängigkeiten zwischen der Ausdehnung der Krone (gemittelt über die acht Richtungen) und dem BHD finden könnte. Die übrigen Parameter (Höhe und Kronenansatz) sollten im korrelativen Zusammenhang zum BHD stehen.

Die Ergebnisse dieser Korrelationsanalysen sind in den Ergebnissen zur Erhebung der Realdaten gezeigt. Es haben sich zwar derartige Zusammenhänge in den Jahren 1996 und 2001 ergeben, bei den Zuwächsen hingegen ließ sich kein Zusammenhang erkennen. Es hatte also keinen Sinn, aus diesen Daten irgendetwas herauslesen zu wollen. Eine Möglichkeit, etwas aus diesen Daten zu machen, war, die Simulation auf der Basis einer Ausgangssituation fortzuschreiben. Die Ausgangssituation ist hier der Zustand von 1996.

Der Zustand zum Zeitpunkt t ist auf der Basis des Zusammenhanges zwischen dem Ausgangsstadium (t_0) und dem jährlichen durchschnittlichem Zuwachs (t_i) berechnet. Die Streuung um diese Regressionsgerade ist sehr groß. Sie ist daher im Modell mitsimuliert worden: Die Residuen sind für alle Zusammenhänge normalverteilt. Die erzeugten jährlichen Zuwächse streuen in der Simulation um die Standardabweichung. Dies sei hier am Beispiel der Durchmesserzuwächse erklärt. Gleichung 1 und Gleichung 2 geben die Gleichungen wieder, mit denen der Durchmesser zum Zeitpunkt t berechnet wurde. Der durchschnittliche jährliche Zuwachs (d_i) berechnet sich aus dem Ausgangsdurchmesser d_0 , der für das Jahr 1996 steht, und einem Korrektursummanden, der die Streuung um die Gerade simuliert (Gleichung 1). Durch Addition des t -fachen jährlichen Zuwachses auf den Ausgangsdurchmesser ergibt sich der Durchmesser zum Zeitpunkt t (Gleichung 2).

$$d_i = 0.021 * d_0 + 0.041 + d_r \quad (1)$$

$$d_t = d_0 + t * d_i \quad (2)$$

5.4 Das L-System

Das L-System komplett darzustellen und im Detail zu erläutern, wäre an dieser Stelle übertrieben, obgleich es nur neun Regeln enthält. Es wird ausführlich im Abschlussbericht dargestellt werden. Hier seien nur die wichtigsten Komponenten erklärt.

Grundlage für die Erstellung des L-Systems ist die zusätzliche Funktionalität von GROGRA hinsichtlich Datenschnittstellen, Sensitivität und stochastischer Modellierung (Kurth und Sloboda (1997, 1999a,b)). Als erstes müssen wir für die Erstellung der Ausgangssituation Rohdaten aus einer Datei einlesen. Dies geschieht in der GROGRA-Spezifikation über eine Arrayvariable. Listing 3 zeigt die Deklaration der Variablen. Die Bedeutung ist: Es soll eine Variable *data* existieren, die ein Array ist. Die Daten stehen in der Datei *system.a01*. Der Array ist zweidimensional und hat dreizehn Spalten und 151 Zeilen. Die Anzahl der Spalten und Zeilen entsprechen genau der Dateigröße.

Listing 3: Deklaration der Arrayvariablen zum Einlesen von Daten

```
\var data array 1 2 13 151,
```

Für die Berechnung des Abstandes zu Nachbarkronen benötigen wir das Mittel der Sensitivität. Listing 4 zeigt die Vereinbarung der Funktion 14 im L-System, die dieses leistet. Diese Codezeile ist die Schnittstelle zwischen regelbasiertem und prozeduralem Modellteil. (Letzterer ist konventionell in der Programmiersprache C implementiert und wird hier nicht weiter erläutert.) Die Variable *dt* bekommt den Abstand zurückgeliefert. Die Null an letzter Stelle der Deklaration bedeutet, dass diese Funktion keinen Übergabeparameter aus dem L-System hat.

Listing 4: Deklaration der Funktion zur Berechnung des Abstandes zur Nachbarkrone.

```
\var dt function 14 0,
```

Für die Simulation der Streuung der jährlichen Zuwächse benötigen wir eine stochastische Variable, die normalverteilt ist. Listing 5 stellt die Deklaration des Streuungsparameters für den Durchmesser dar. Die Variable *dres* ist eine normalverteilte Zufallsvariable, mit dem Mittelwert Null mit einer Varianz 0.0289.

Listing 5: Deklaration einer normalverteilten Zufallsvariablen

```
\var dres normal 0 0.0289,
```

Zur Verdeutlichung der angewendeten Regeln seien hier zwei Regeln als Beispiel für das Wachstum der Kronenausdehnung herausgegriffen (vgl. Listing 6). In Zeile zwei kommt die Funktion 14 zum Einsatz. Ist der Abstand größer als unser Minimalabstand aus den Realdaten (vgl. Abbildung 9), dann findet ein Wachstum der Kronenausdehnung statt. Ist der Abstand kleiner, tritt die Regel aus Zeile 2 in Kraft: Das Wachstum stagniert. Die deklarierten Variablen im Symbol *crbase* sind:

- *rb* = Ausgangsradius,
- *rr* = normalverteilte Zufallsvariable,
- *ri* = jährlicher Zuwachs,
- *t* = Zeitpunkt.

Von diesen Variablen verändert sich nur t . Die Zeit wird in Jahresschritten heraufgezählt. ri wird zu jedem Schritt neu aus den Korrelationsparametern zu rb und dem Zufallssummand rr berechnet.

Der eigentliche Zuwachs erfolgt in der interpretativen Regel in Zeile 3. Hier wird an letzter Stelle die Gleichung 2 äquivalent für die Kronenradien angewendet. Das F -Kommando erzeugt eine Kronenbasiseinheit mit der Länge l_t .

Vorher wird der reinen Übersicht halber die Farbe auf 10 (rot) gesetzt, ein virtueller Durchmesser erzeugt und als Blattparameter (Befehl N) eine 1 definiert. Der Blattparameter wird dazu benutzt, um diese Kronenbasiseinheit von anderen Einheiten zu unterscheiden.

Listing 6: Die Regeln zum Wachstum der Kronenausdehnung.

```
( dtt > (-1.5) ) crbase ( rb , rr , ri , t ) # crbase ( rb , rr , ( rb * (-0.0541) + 0.201 + rr ) , t -
crbase ( rb , rr , ri , t ) # crbase ( rb , rr , ri , t + 1 ) ,
crbase ( rb , rr , ri , t ) ## P10 D1 ( 0.01 ) N ( 1 ) F ( rb + t * ri )
```

5.5 Ergebnisse

Das Modell ist auf dem Rechner über sechs Jahre in das Jahr 2001 simuliert worden. Abbildung 11 zeigt das Ergebnis grafisch für das Ergebnisjahr. Interessant war, wie sich die Simulation von den Realwerten unterscheiden würde. Für den Bericht seien hier kurz die Durchmesserverteilungen wiedergegeben. Abbildung 12 zeigt die Verteilung für den Realbestand (vgl. auch Tabelle 3), Abbildung 13 für die Simulation. Der durchschnittliche BHD liegt in der Simulation niedriger. Tabelle 5 zeigt nochmal die Werte der Simulation für die Höhe, den Kronenansatz und den durchschnittlichen Kronenradius.

5.6 Diskussion

Die Form des Modelles ist absichtlich sehr einfach gewählt: Es bringt per se nur die Information, wie der Bestand durchschnittlich zuwächst. Das Datenmaterial hat aber tiefergehende Analysen und Interpretationen nicht zugelassen. Ein genereller Zusammenhang zwischen Zuwächsen und BHD hat sich nicht bestätigt.

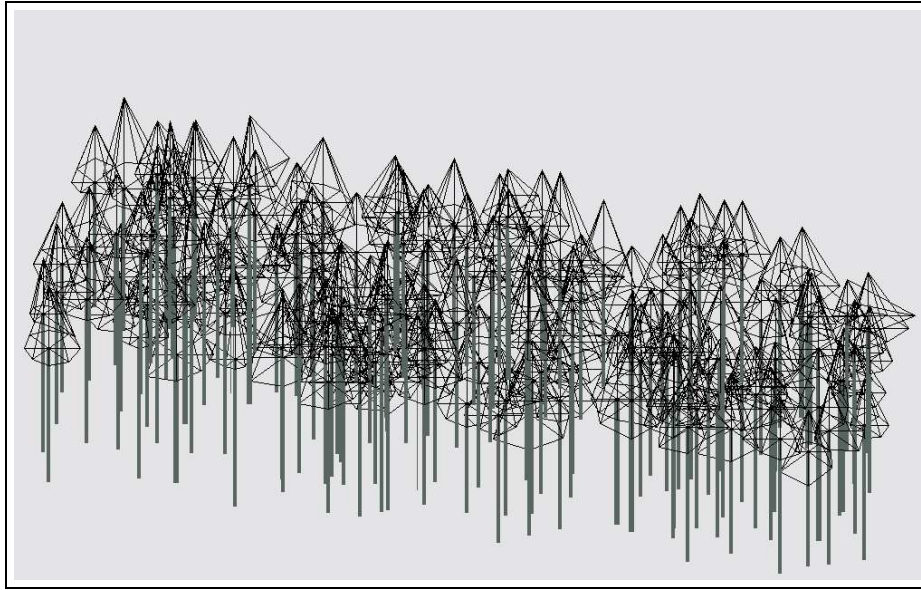


Abbildung 11: Der simulierte Bestand nach 6 Jahren (2001).

Die Simulation hat in den Ergebnissen ähnliche Werte erzeugt, wie die Realmessungen ergeben haben. Die einfache Form des Modelles und der sehr kurze Simulationszeitraum von sechs Jahren ließ das aber auch erwarten. Auffällig ist der kleinere Durchmesser in der Simulation. Er lässt sich aber durch die Durchforstungsmaßnahme begründen, die der Realbestand erfahren hat. In einer Durchforstung werden im Schnitt Bäume mit geringeren Durchmessern entnommen. In der Simulation ist dieser Vorgang aber nicht abgebildet, so dass der Durchschnitt in der Simulation niedriger liegen muss.

Insgesamt hat sich gezeigt, dass es außerordentlich schwierig ist, aus derartigem Rohdatenmaterial Modelle in dieser aggregierten Form zu entwickeln. Das Modell, wie es hier gezeigt wurde, simuliert nur den Bestand in Syke. Schon der Nachbarbestand kann anders reagieren.

Einen potenziellen Ausweg könnten Modelle liefern, die näher an den eigentlichen Kausalitäten liegen. Da derartige Modelle in der Regel sehr rechenintensiv sind, ist die Aggregation derartiger Modelle wahrscheinlich ein guter Weg, einen größeren Grad an Allgemeinheit zu erhalten. Es ist daher vorgesehen, ein Bestandesmodell auf der Grundlage einer Vereinfachung von LIGNUM (Perttunen et al., 1998) zu erstellen.

Variable	N	Mean	Std Dev	Minimum	Maximum
BHD	151	20.7	4.8	10.3	34.9
Höhe	151	19.5	2.4	13.3	25.4
KA	151	13.6	1.3	10.2	16.9
KRMEAN	151	2.1	0.4	0.8	2.3

Tabelle 5: Teilbestandeskennwerte 2001 simuliert. KA = Kronenansatz, KRMEAN = Mittlerer Kronenradius pro Baum, DMEAN = Mittlerer Abstand zum nächsten Nachbarn pro Baum

6 Modellaggregation einer LIGNUM-Kiefer

6.1 Material und Methoden

Dieses Kapitel trägt den Namen Material und Methoden, obgleich das „Material“ eigentlich virtuell ist. Das Material sind in diesem Fall drei simulierte LIGNUM-Kiefern. Das Modell LIGNUM als solches ist schon beschrieben. Es erzeugt eine dreidimensionale Kronenstruktur einer Kiefer auf der Grundlage des Lichtgenusses und der damit verbundenen Assimilationsrate.

Die Kiefern, die hier zu Grunde liegen, sind über 50 Jahre simuliert worden. Die Jahreszahl sollte den oben beschriebenen Realdaten entsprechen. Die Kiefer hat neben der internen Lichtkonkurrenz nicht den vollen Lichtgenuss erhalten: Der Lichteinfall von außen ist durch einen hypothetischen Bestand reduziert worden. Die Reduktionsrate hing dabei von der Bestandesdichte im den Baum umgebenden Bestand ab. In diesem Fall betrug die Dichte 1500, 2000 und 5000 Bäume pro Hektar (3 Varianten).

Die virtuellen Bäume sind ebenfalls einer Datenanalyse unterzogen worden. Dabei kam die Datenschnittstelle zwischen GRODISC und LIGNUM zum Einsatz. Die Analyse differierte von der der Realdaten. Der Grund liegt auf der Hand: Wir hatten hier nur drei Bäume zur Verfügung, die miteinander nichts zu tun hatten. Andererseits lagen mehr Informationen über die zeitliche Entwicklung der Baumparameter BHD, Höhe, Kronenansatz und Kronenausdehnung vor. LIGNUM hat dazu die geometrische Struktur

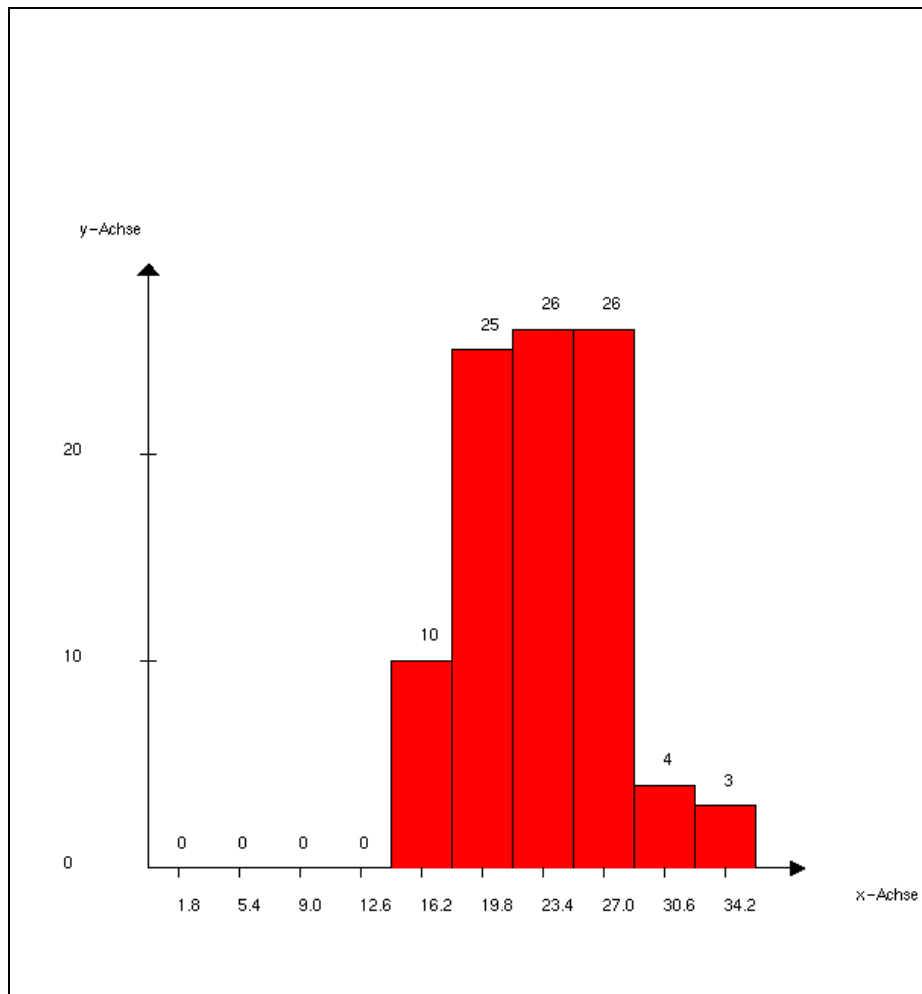


Abbildung 12: Die Durchmesserverteilung des realen Bestandes im Jahre 2001.

eines jeden Jahresschrittes zurückgegeben. Somit lagen 50 Einzelwerte pro Baum für die gleichen Parameter vor, die schon in der Analyse des Realbestandes gemessen wurden.

Die Baumparameter sind von GRODISC bestimmt worden. GRODISC hat dazu die Kronenstruktur, die jedes einzelne Segment enthält, in eine vereinfachte Krone umgerechnet. Es ist dabei die Messung in Form einer Ablotung simuliert worden. Die Höhe des Baumes entsprach dem größten Z-Wert einer Einheit im Baum, der BHD dem Durchmesser derjenigen Einheit, die eine Höhe von 1.3 Meter umschließt. Der Kronenansatz schließlich wurde analog zur Realmessung von derjenigen Achse bestimmt, die von der Ordnung eins ist und noch Blattmasse enthielt.

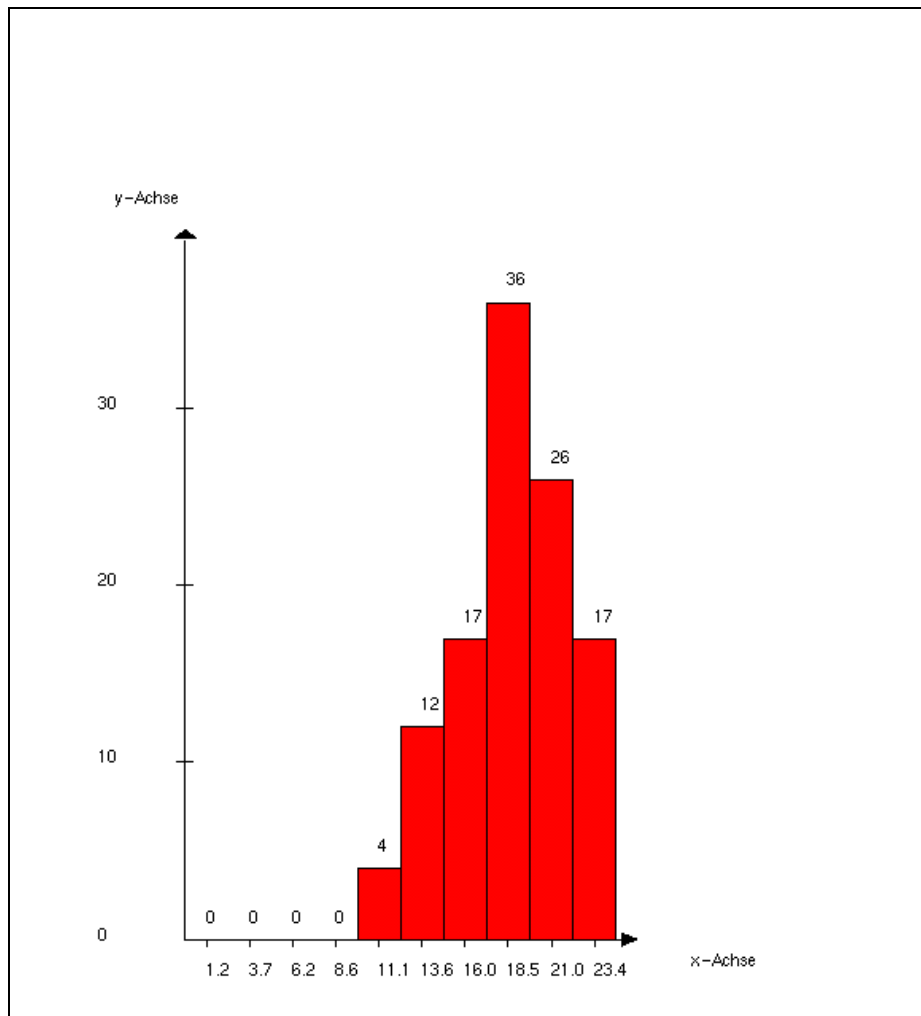


Abbildung 13: Die Durchmesserverteilung des simulierten Bestandes im Jahre 2001.

6.2 Ergebnisse

Die Simulationsläufe sind mit einer Bestandesdichte von 1500, 2000 und 5000 Bäumen pro Hektar durchgeführt worden. Zuallererst die zeitliche Entwicklung der Parameter: Die Abbildungen 14, 15 und 16 zeigen den Entwicklungsgang. Die Kronenausdehnung ist auch hier über die acht Himmelsrichtungen gemittelt dargestellt.

Der Kronenansatz ist hier nur für denjenigen Baum dargestellt, der unter einer Bestandesdichte von 5000 Bäumen pro Hektar zu „leiden“ hatte (Abbildung 17). Alle zeitlichen Entwicklungen dieses Parameters hatten die gleiche Form eines etwas zufällig anmutenden Hin- und Herspringens.

Die Durchmesserentwicklung zeigt einen seltsamen Gang: Der Zuwachs bricht bei 22 Jahren ab. Es konnte dafür bis jetzt keine Erklärung gefunden werden. Man muss allerdings dazu feststellen, dass die Größen der Durchmesser durchaus realistisch erscheinen und auch in einem normalen Verhältnis zur Höhe stehen. Die Höhenentwicklung verläuft interessanterweise anders — sie stoppt nicht in ihrem Wachstum und zeigt auch hier ein Verhalten, was man durchaus in der Realität antreffen kann.

Die Ergebnisse bei den Kronenansätzen zeigen ein munteres Hin- und Herspringen. Er macht sehr lange das obere Ende der ersten Einheit aus, springt dann aber fast zufällig nach oben und nach unten.

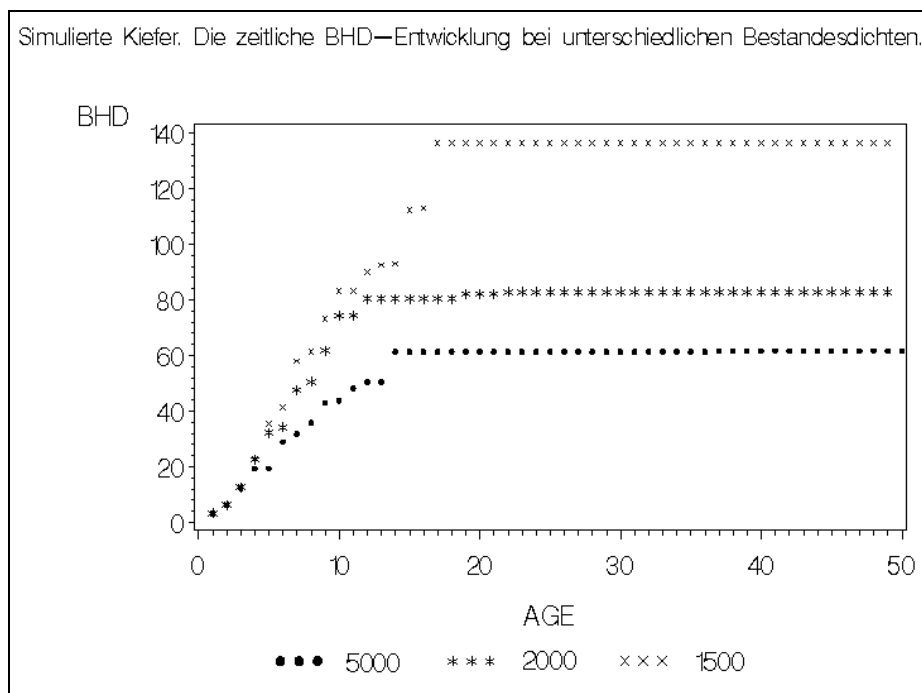


Abbildung 14: Die Entwicklung der Durchmesser in der Zeit bei verschiedenen Bestandesdichten.

6.3 Diskussion

Es hat sich beim Aggregationsansatz gezeigt, dass die LIGNUM-Kiefer in den zusammengefassten Daten BHD, Höhe, Kronenansatz und Kronenausdehnung Ergebnisse liefert, die der Realität durchaus nahe kommen. Dass es nicht genau passt, ist nicht weiter verwun-

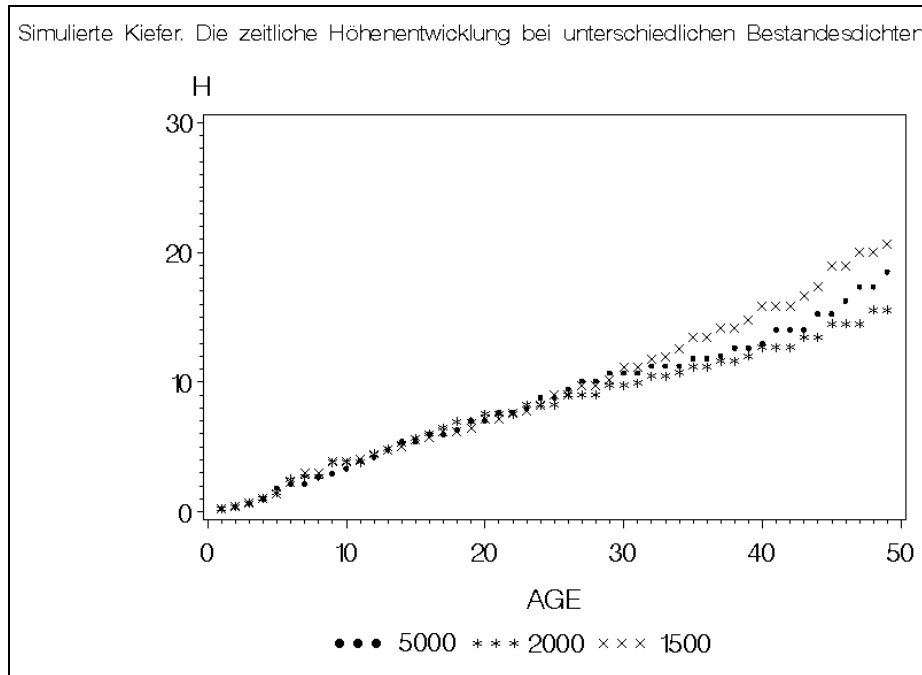


Abbildung 15: Die Entwicklung der Höhen in der Zeit bei verschiedenen Bestandesdichten.

derlich. Als einziger Standortsfaktor ist bei der Kiefer das Lichtregime berücksichtigt. Es gibt sicherlich auch noch andere Faktoren, die das Wachstum einer Kiefer beeinflussen.

Trotz alledem haben sich in der Entwicklung der Parameter in der Zeit einige Unzulänglichkeiten des Modells gezeigt. Es ist seltsam, dass das Durchmesserwachstum stagniert. Die Kohlenstoffallokationsbedingungen im Stamm sollten ein Durchmesserwachstum bewirken. In dieser Frage ist Klärungsbedarf. Aus den gewonnenen Daten lässt sich nicht ablesen, ob ein Fehler in der Implementierung des Modells vorliegt oder ob GRODISC die Durchmesser fehlerhaft analysiert hat. Logisch ist das Verhältnis zwischen den einzelnen Lichtregimen, das durch die Bestandesdichte geregelt wird. Je höher die Bestandesdichte, desto niedriger der Zuwachs.

Bei den Höhen hingegen zeigt sich kein derartiger Zusammenhang. Dies ist aber auch in der Natur beobachtbar. Ganz im Gegenteil, manche Bäume haben sogar ein verstärktes Höhenwachstum in dichteren Beständen.

Der Kronenansatz ist hier nicht auf diese Art und Weise betrachtet worden. Durch das starke Hin- und Herspringen im zeitlichen Ablauf wäre eine Aussage im Verhält-

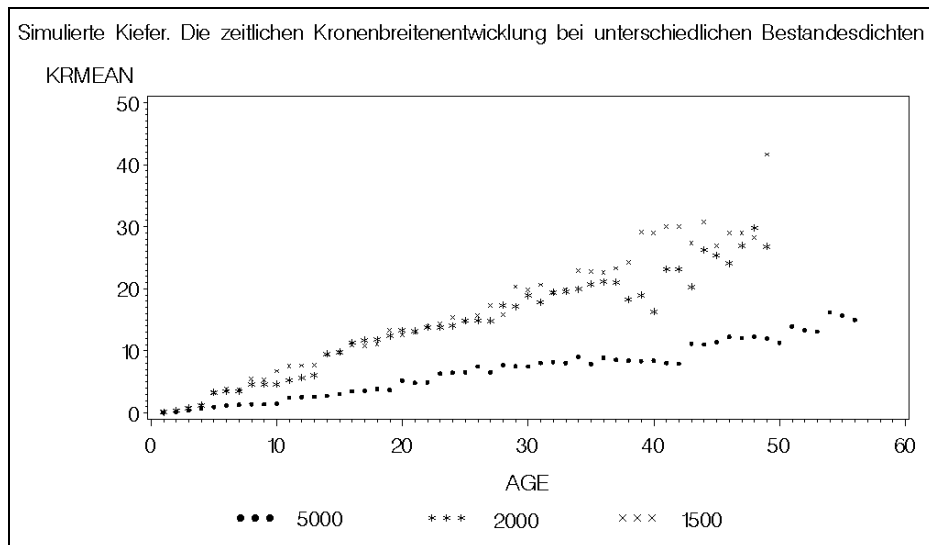


Abbildung 16: Die Entwicklung der durchschnittlichen Kronenbreiten in der Zeit bei verschiedenen Bestandesdichten.

nis zur Bestandesdichte nicht möglich gewesen. Hier besteht sicherlich noch erheblicher Klärungsbedarf. Es ist nicht ersichtlich, warum eine Achse plötzlich wieder Blattmasse besitzt, wenn sie vorher „dem Tode geweiht“ war.

Die Kronenausdehnung reagiert nun wieder in erwarteter Weise. Sie ist bei großen Dichten deutlich geringer. LIGNUM scheint tatsächlich auf den geringeren Lichtgenuss mit geringeren Zuwachsraten bei den Längen der Wachstumseinheiten zu reagieren. Etwas anderes wäre auch seltsam gewesen, ist doch die Länge einer Wachstumseinheit abhängig von den umgesetzten virtuellen Assimilaten (Perttunen et al., 1998).

7 Kooperation und Präsentation der Ergebnisse

Die Ergebnisse des Projektes sind auf Tagungen und Meetings präsentiert worden. National existiert ein jährlich stattfindender Workshop mit dem Titel „Workshop on Individual-based Models and Structural-Functional Models in Ecology“. Auf dem siebten und achten Treffen sind Ergebnisse in Form von Vorträgen präsentiert worden. Zum achten Workshop ist zusätzlich ein Tagungsband entstanden, in dem wir veröffentlicht haben (Dzierzon und Kurth, 2001a).

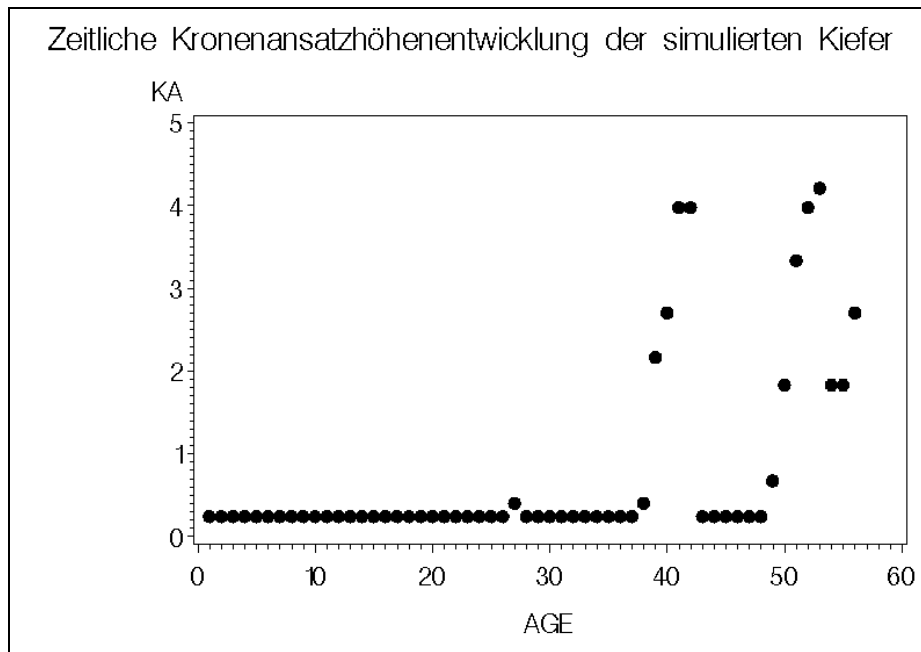


Abbildung 17: Die Entwicklung des Kronenansatzes in der Zeit bei einer Bestandesdichte von 5000 Bäumen pro Hektar.

International waren wir auf der Tagung „Third International Workshop on Functional-Structural Tree and Stand Models“ (Montréal, Kanada) im Oktober 2001 mit einem Vortrag vertreten. Auch zu dieser Tagung ist eine Veröffentlichung geplant, im *Canadian Journal of Forest Science*.

Es ist vorgesehen, abschließende Ergebnisse auf dem Meeting „Hierarchical treatment of multi-scale processes in tree and stand models“ an der Universität Helsinki (26.–27. 2. 2002) zu präsentieren.

Die Arbeitsgruppe Pflanzenmodellierung am Institut für Forstliche Biometrie und Informatik der Universität Göttingen, an dem dieses Projekt angesiedelt ist, hat eine Vielzahl von nationalen und internationalen Kooperationspartnern. Mit mehreren dieser Partner ist die Zusammenarbeit im Vorfeld und im Laufe dieses Projektes vertieft worden. So hat der Bearbeiter einleitend einen viermonatigen Auslandsaufenthalt an der Finnischen Forstlichen Versuchsanstalt (METLA) in Vantaa und an der Universität Helsinki geleistet. Die Ergebnisse dieses Aufenthaltes sind in dieses Projekt eingeflossen. Im Zuge dieser Kooperation ist eine Veröffentlichung in *Silva Fennica* entstanden (Dzierzon et al.,

2001b). Der Verfasser wird in diesem Jahr erneut die METLA aufsuchen, um die durch die Zusammenarbeit entstandenen Ergebnisse abschließend zu beraten.

Das CIRAD in Montpellier, das schon lange Kooperationspartner unserer Arbeitsgruppe ist, hat dazu beigetragen, dass die Publikationen Dzierzon und Kurth (2001a), Knauff et al. (2001) und die Dissertation von Knauff (2000) entstanden. Das CIRAD hat die Software AMAPmod zur Verfügung gestellt und uns bei Fragen der Anwendung unterstützt.

Zu nennen ist auch hier die Niedersächsische Forstliche Versuchsanstalt Göttingen, die freundlicherweise Daten und Messgeräte zur Verfügung gestellt hat.

Wichtig für unsere Arbeit ist die Zusammenarbeit mit dem „Laboratoire de Recherches en Sciences Forestières, Unité Ecosystèmes Forestiers et Dynamique du Paysage“ in Nancy (ENGREF), Frankreich. Sie haben wichtige Daten bereitgestellt. Wir haben im Gegenzug Software erstellt, die in der Dissertation von Duchiron (2000) benutzt wurde. Bei der Verteidigung der Dissertation an der Universität Nancy war der Bearbeiter des Projektes als eingeladener Sachverständiger anwesend.

Literatur

- de Reffye, Ph., Fourcaud, Th., Blaise, F., Barthélémy, D., and Houllier, F. 1997. A functional model of tree growth and tree architecture. *Silva Fennica* 31: 297-311.
- Deussen, O., Hanrahan, P., Lintermann, B., Mech, R., Pharr, M., and Prusinkiewicz, P. 1998. Realistic modeling and rendering of plant ecosystems. In: *Proc. SIGGRAPH 98, Computer Graphics Proceedings, Ann. Conf. Ser.* 1998: 275-286.
- Duchiron, M.-S. 2000. Analyse de la structure des forêts irrégulières et mélangées sur les landes sableuses non hydromorphes de l'Allemagne du nord. Thèse présentée à l'École du Génie Rural, des Eaux et des Forêts, Nancy, France.
- Dzierzon, H. und Kurth, W. 2001. LIGNUM: A Finnish tree growth model and its interface to the French AMAPmod database. In Hölker, F. and Breckling, B. (eds.), *Proc. 8th workshop Individual-based models and structural-functional models, Helmenau Germany (10.-12. 7. 2000)*. Peter Lang Verlag, Frankfurt, Germany (in press).
- Dzierzon, H., Perttunen, J., Kurth, W., Sievänen, R., Sloboda, B. 2001. Enhanced possibilities for analyzing tree structure as provided by an interface between different modelling systems. *Silva Fennica* (submitted).
- Eschenbach, Ch. 2000. The effect of light acclimation of single leaves on whole tree growth and competition – an application of the tree growth model ALMIS. *Ann. For. Sci.* 57: 599-609.
- Füldner, K. (1995). Strukturbeschreibung von Buchen-Edellaubholz-Mischwäldern. *Cuvillier, Göttingen* 146. Zugl.: Göttingen, Univ., Forstwiss. Fachber., Diss., 1995.
- Godin, C. 2000. Representing and encoding plant architecture: A review. *Ann. For. Sci.* 57: 413-438.
- Godin, C., und Caraglio, Y. 1998. A multiscale model of plant topological structures. *J. Theor. Biol.* 191: 1-46.

- Godin, C., Guedon, Y., und Costes, E. 1998. Exploration of a plant architecture database with the AMAPmod software illustrated on an apple tree hybrid family. *Agronomie* 19: 163-184.
- Guericke, M. (2001): Untersuchungen zur Wachstumsdynamik von Mischbeständen aus Buche und Europ. Lärche (*Larix decidua*, Mill) als Grundlage für ein abstandsabhängiges Einzelbaumwachstumsmodell. Dissertation, Universität Göttingen. Cuvillier, Göttingen.
- Knauff, F.-J. (2000). Entwicklung von Methoden zur GIS-gestützten Visualisierung von Waldentwicklungsszenarien. Dissertation an der Fakultät für Forstwissenschaft und Waldökologie, Universität Göttingen.
- Knauff, F.-J., Sloboda, B., und Kurth, W. (2001). Modellintegration bei der Erzeugung virtueller Waldlandschaften. *Allgemeine Forst- und Jagdzeitung*, 172 (8/9): 161-168.
- Kramer, H. (1984): Grundlagen zur forstlichen Ertragskunde, 3. Aufl., Göttingen.
- Kurth, W. 1994. Growth Grammar Interpreter GROGRA 2.4. - A software tool for the 3-dimensional interpretation of stochastic, sensitive growth grammars in the context of plant modelling. Introduction and reference manual. In: *Berichte des Forschungszentrums Waldökosysteme*, Ser. B, Bd. 38. Göttingen, Germany.
- Kurth, W. 1998. Die Simulation der Baumarchitektur mit Wachstumsgrammatiken. Wissenschaftlicher Verlag, Berlin, Germany.
- Kurth, W., und Sloboda, B. 1997. Growth grammars simulating trees - an extension of L-Systems incorporating local variables and sensitivity. *Silva Fennica*, 31: 285-295.
- Kurth, W., und Sloboda, B. (1999). Tree and stand architecture and growth described by formal grammars - I. Non-sensitive trees. *J. For. Sci.*, 45: 16-30.
- Kurth, W., und Sloboda, B. (1999). Tree and stand architecture and growth described by formal grammars - II. Sensitive trees and competition. *J. For. Sci.*, 45: 53-63.

- List, R., Küppers, M., und Schröder, F. 1994. Madeira – ein Programm zur Simulation des Wachstums (Biomasse und Architektur) von Holzgewächsen. Verhandlungen der Gesellschaft für Ökologie, 23, 23. Jahrestagung der Gesellschaft für Ökologie, Innsbruck, 6.-9. 9. 1993, 39-48.
- Nagel, J., Albert, M., Schmidt, M. (2000): BWINPro Programm zur Bestandesanalyse und Prognose. Handbuch zur Version 5.1. Niedersächsische Forstliche Versuchsanstalt, Göttingen.
- Niklas, K. J. (1994): Plant allometry. The scaling of form and process. UCP, Chicago / London.
- Oppelt, A.L., Kurth, W., Dzierzon, H., Jentschke, G. und Godbold, D.L. (2000). Structure and fractal dimensions of root systems of four co-occurring fruit tree species from Botswana. *Ann. For. Sci.* 57(5/6): 463-475.
- Perttunen, J., Sievänen, R., Nikinmaa, E., Salminen, H., Saarenmaa, H. und Väkevä, J. (1996). LIGNUM: A tree model based on simple structural units. *Ann. Bot.* 77: 87-98.
- Perttunen, J., Sievänen, R. und Nikinmaa, E. 1998. LIGNUM: a model combining the structure and the functioning of trees. *Ecological Modelling* 108: 189-198.
- Pfreundt, J. (1988). Modellierung der räumlichen Verteilung von Strahlung, Photosynthesekapazität und Produktion in einem Fichtenbestand und ihrer Beziehung zur Bestandesstruktur. In: *Berichte des Forschungszentrums Waldökosysteme, Ser. A, Bd. 39.* Göttingen.
- Pfreundt, J., und Sloboda, B. (1996). The relation of local stand structure to photosynthetic capacity in a spruce stand: a model calculation. *Lesnictví-Forestry*, 42: 149-160.
- Pretzsch, H. 1992. Modellierung der Kronenkonkurrenz von Fichte und Buche in Rein- und Mischbeständen. *Allgemeine Forst- und Jagdzeitung*, 163 (11/12): 203-213.
- Prodan, M. (1965): *Holzmesslehre.* J.D. Sauerländer's, Frankfurt.

Prusinkiewicz, P., und Lindenmayer A. (1990): The algorithmic beauty of plants. Springer, New York.

Sloboda, B., und Pfreundt, J. (1989). Tree and stand growth. A system analytical spatial model with consequences for test planning for thinning and single tree development. Artificial intelligence and growth models for forest management decisions, Publication No. FWS-1-89, Blacksburg, Virginia.

Stroustrup, B. 1997. Die C++ Programmiersprache. Addison-Wesley, Bonn. 3. Auflage.