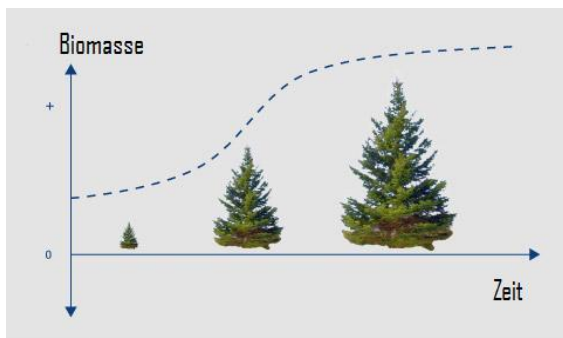
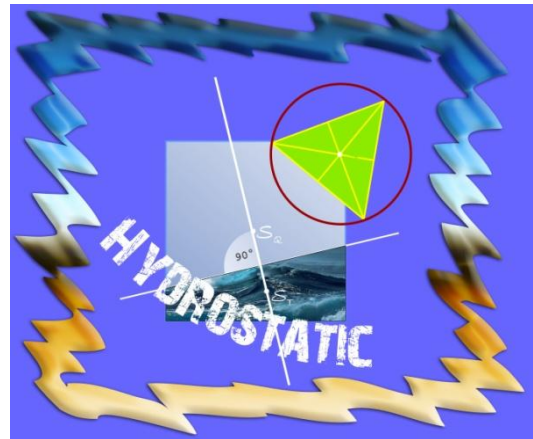
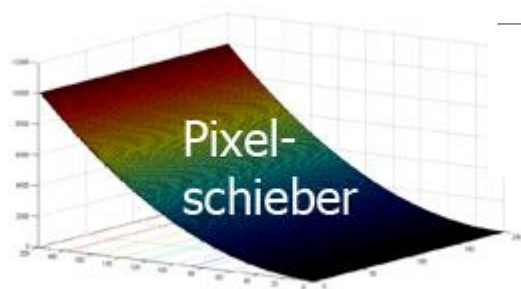


WOCHE DER MODELLIERUNG MIT MATHEMATIK

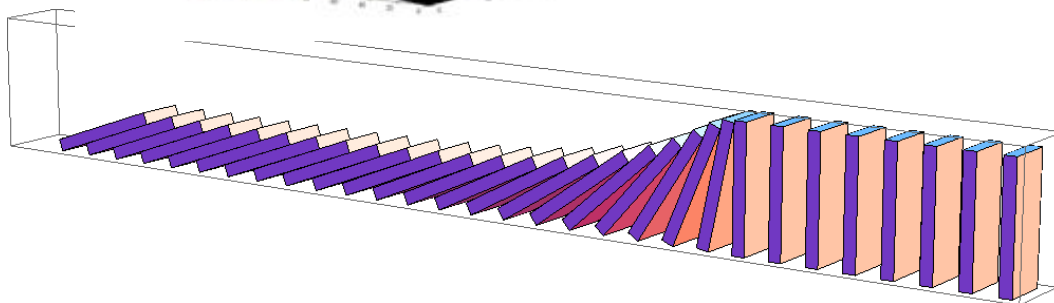
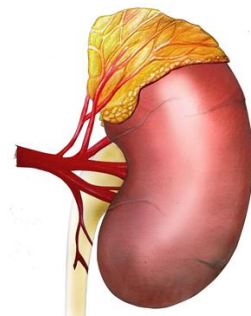


DOKUMENTATIONS- BROSCHÜRE

7.2. – 13.2.2010



$$\frac{dM}{dt}_B =$$



WOCHE DER MODELLIERUNG MIT MATHEMATIK



Pöllau bei Hartberg, 7.2. – 13.2.2010

Weitere Informationen:

<http://math.uni-graz.at/modellwoche/2010/>



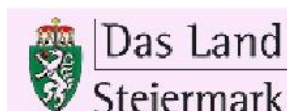
Vorwort

Viele Wissenschaften erleben zurzeit einen ungeheuren Schub der Mathematisierung. Mathematische Modelle, die vor wenigen Jahrzehnten noch rein akademischen Wert hatten, können heute mit Hilfe von Computern vollständig durchgerechnet werden und liefern praktische Vorhersagen, die helfen, Phänomene zu verstehen, Vorgänge zu planen, Kosten einzusparen. Damit unsere Gesellschaft auch in Zukunft mit der technologischen Entwicklung schritthält, ist es wichtig, bereits junge Leute für diese Art mathematischen Denkens zu begeistern und in der Gesellschaft das Bewusstsein für den Nutzen angewandter Mathematik zu heben. Dies war für uns einer der Gründe, die Woche der Modellierung mit Mathematik zu veranstalten.

Nun ist leider für viele Menschen Mathematik ein Schulfach, mit dem sie eher unangenehme Erinnerungen verbinden. Umso erstaunlicher erscheint es, dass Schülerinnen und Schüler sich freiwillig melden, um eine ganze Woche lang mathematische Probleme zu wälzen - und dabei auch noch Spaß haben. Sie erleben hier offensichtlich die Mathematik auf eine Art und Weise, wie sie der Schulunterricht nicht vermitteln kann. Die jungen Leute arbeiten und forschen in kleinen Gruppen mit Wissenschaftler/innen an realen Problemen aus den verschiedensten Bereichen und versuchen, mit Hilfe mathematischer Modelle neue Erkenntnisse zu gewinnen. Sie arbeiten ohne Leistungsdruck, dafür mit Eifer und Enthusiasmus, rechnen, diskutieren, recherchieren, oft auch noch am späten Abend, in einer entspannten und kreativen Umgebung, die den Schüler/innen und betreuenden Wissenschaftler/innen gleichermaßen Spaß macht. Als Projektbetreuer konnte ich auch in diesem Jahr wieder erleben, wie eigenes Entdecken und Selbstmotivation das Verhalten der Schüler/innen während der ganzen Modellierungswoche bestimmen. Sie lernen eine Arbeitsmethode kennen, die in beinahe allen Details den Arbeitsmethoden einer Forschergruppe entspricht. Bei keiner anderen Gelegenheit erfahren Schüler/innen so viel über Forschung wie bei so einer Veranstaltung.

Modellierungswochen gab bzw. gibt es zum Beispiel auch in den USA, in Deutschland, in Italien. Wir verdanken Herrn Prof. Dr. Stephen Keeling den Vorschlag, auch durch die Universität Graz so eine Woche zu veranstalten, und seiner unermüdlichen Organisationsarbeit das tatsächliche Zustandekommen. Er leitet nun bereits zum sechsten Mal diese inzwischen zur Institution gewordene Veranstaltung. Ihm sei an dieser Stelle noch einmal ausdrücklich und herzlich gedankt. Besonders wichtig war in den vergangenen Jahren auch die Unterstützung durch den langjährigen Mentor der Modellierungswoche, Herrn o.Univ.-Prof. Dr. Franz Kappel, der oft auch eine eigene Gruppe mit interessanten Problemstellungen betreut hat.

Wir danken dem Landesschulrat für Steiermark, und hier insbesondere Frau Landesschulinspektorin Frau HR Mag. Marlies Liebscher, für die Hilfe bei der Organisation und ihre kontinuierliche Unterstützung der Idee einer Modellierungswoche. Ohne den idealistischen, unentgeltlichen und engagierten Einsatz der direkten Projektbetreuer Dr. Jerry Batzel, Dr. Stephen Keeling, Dr. Georg Propst und Dr. Peter Schöpf – alle Institut für Mathematik und Wissenschaftliches Rechnen – hätte diese Modellierungswoche nicht stattfinden können.



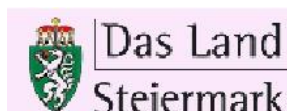
Besonderer Dank gebührt ferner Herrn Mag. Christoph Gruber, der die ganze Veranstaltung betreut und auch die Gestaltung dieses Berichtes übernommen hat, Frau Gerlinde Krois für die tatkräftige Hilfe bei der organisatorischen Vorbereitung, und Herrn Dr. Georg Propst für die Hilfe bei der Betreuung der Hard- und Software.

Erstmalig wird heuer die Modellierungswoche selbst wissenschaftlich untersucht. Die fachdidaktische Begleitforschung wird von Herrn Mag. Christoph Gruber im Rahmen einer Dissertation an unserem Institut und im Auftrag des Bundesministeriums für Wissenschaft und Forschung durchgeführt. Wir danken den Schülerinnen und Schülern für ihre Geduld beim Ausfüllen zahlreicher Fragebögen und für ihre Bereitschaft, sich psychologischen Tests aller Art zu unterziehen.

Finanzielle Unterstützung erhielten wir vom Land Steiermark durch Landesrätin Mag. Kristina Edlinger-Ploder und von der Karl-Franzens-Universität Graz durch Vizerektor Prof. Dr. Martin Polaschek, Dekan Prof. Dr. Karl Crailsheim und Prof. Dr. Karl Kunisch, dem Leiter des Instituts für Mathematik und Wissenschaftliches Rechnen.

Pöllau, am 13. Februar 2010

Bernd Thaller
Institut für Mathematik und Wissenschaftliches Rechnen
Karl-Franzens-Universität Graz



Mathematik-Modellierungswoche

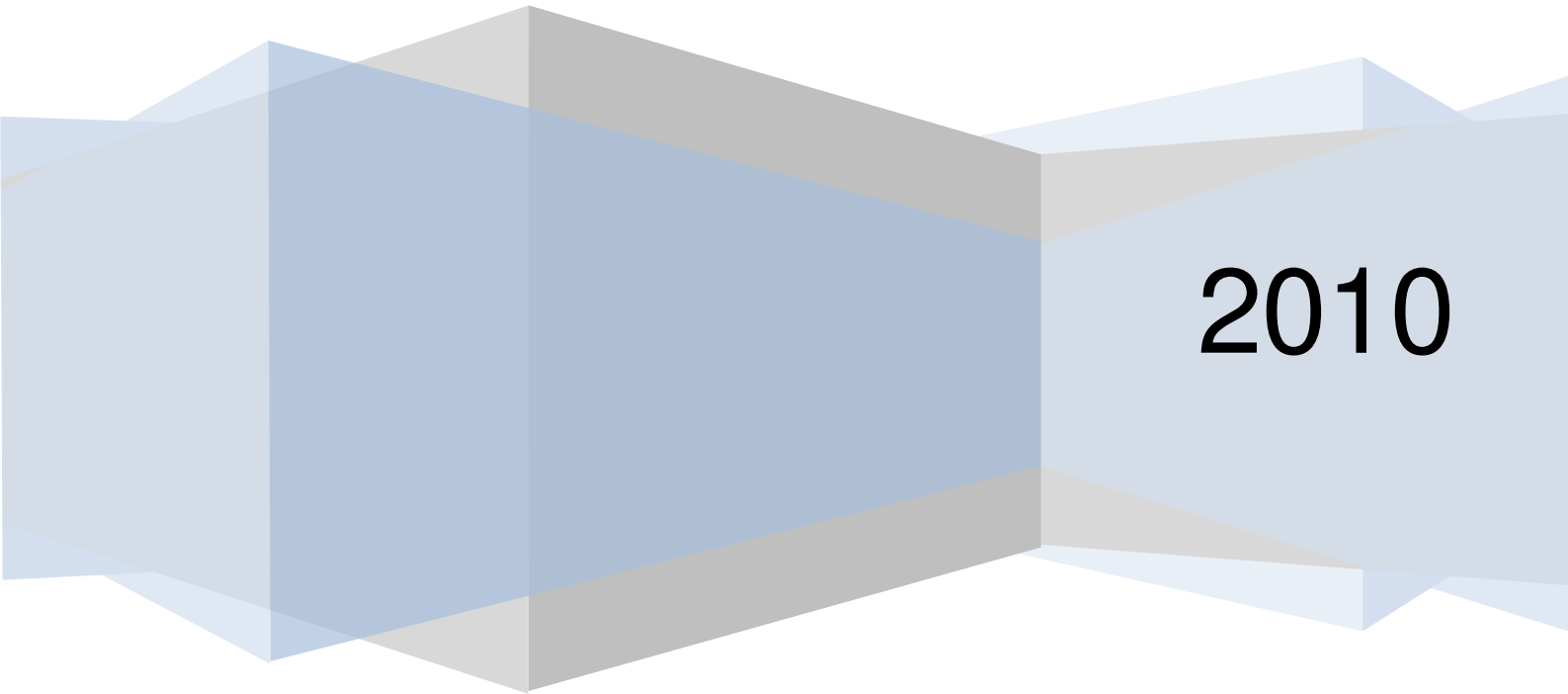
Physiologie

Komplikationen bei der Hämodialyse

Leiter: Prof. Jerry Joseph Batzel

Modellierer:

- Philipp Gabler
- Nikolaus von Niessen
- Robert Schwarzl
- Philipp Schröttner
- Christopher Tscherne
- Katharina Tiran



2010

Inhaltsverzeichnis

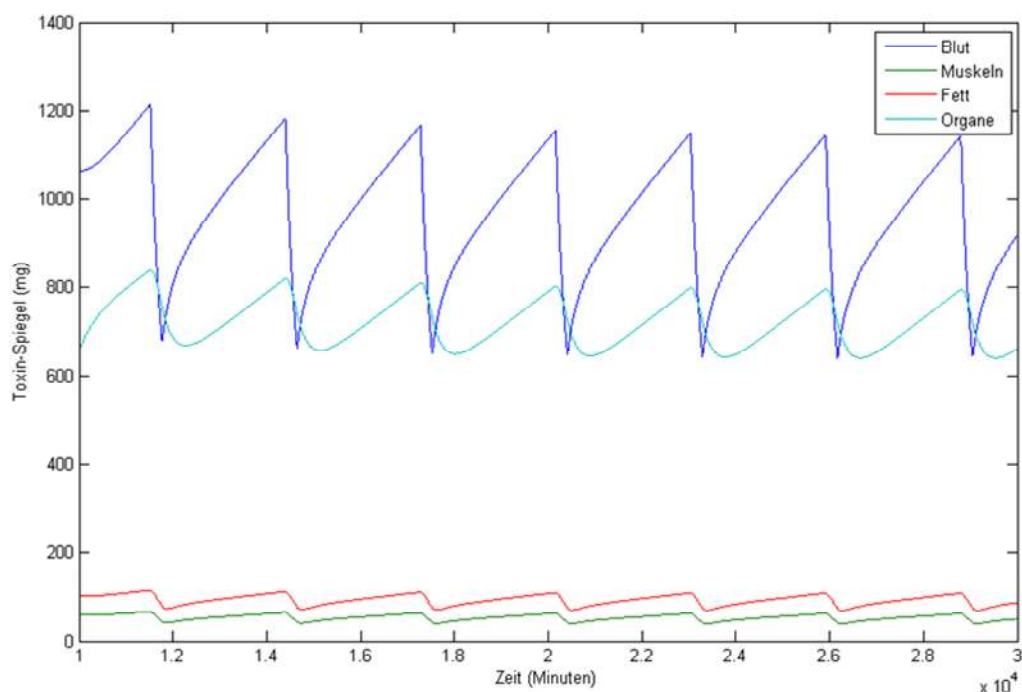
Einleitung.....	3
Die Niere.....	4
Modellierung der Dialysebehandlung	5
Tägliche Dialyse (1440 min):.....	5
Dialyse nach 2 Tagen (2880 min):	6
Dialyse nach 3 Tagen (4320 min):	6
Die Hämodialyse.....	7
Annahmen für die effektiven Volumen.....	8
Sonstige Konstanten.....	8
Kompartimentenmodell	9
Kompartimentanalyse für Hämodialyse.....	9
FMS-Theorie	11
(FAT-MEN-SURVIVING).....	11
Beweis	12
Matlab	18
Variablen	18
Matrizen	18
Die Schleife.....	18
Die for-Schleife:	18
Die while-Schleife:	18
Die Abfrage.....	19
Die if-Abfrage:	19
Zeichenfunktionen.....	19
Handhabung von Bildern.....	19

Einleitung

In diesem Projekt erforschen wir den Vorgang der Dialyse, die im Groben die Funktion der Nieren übernimmt. Die Überlebensquote unter Dialysepatienten ist leider relativ niedrig: es gibt einfach zu viele Nebenwirkungen. Viele davon sind mathematisch erklärbar; beispielsweise arbeiten Nieren rund um die Uhr, aber niemand hängt die ganze Zeit am Dialysegerät. In der Zwischenzeit kann die Toxinkonzentration im Blut erheblich steigen, oftmals verdoppelt sie sich bis zur nächsten Sitzung. Mit der Aufstellung unseres Modells versuchen wir genau das und vieles Andere zu bestimmen.

In mathematischen Modellen können Konzentrationen, absolute Massen und viele Faktoren simuliert werden, von denen das Verhalten während und nach der Behandlung abhängt. Zur Vermeidung von höherer Mathematik erzeugten wir keine hundert-prozentigen Formeln, sondern die Ergebnisse wurden mit MatLab® näherungsweise Graphen bestimmt, mit denen der Verlauf der Giftkonzentrationen bei Patienten dargestellt werden kann.

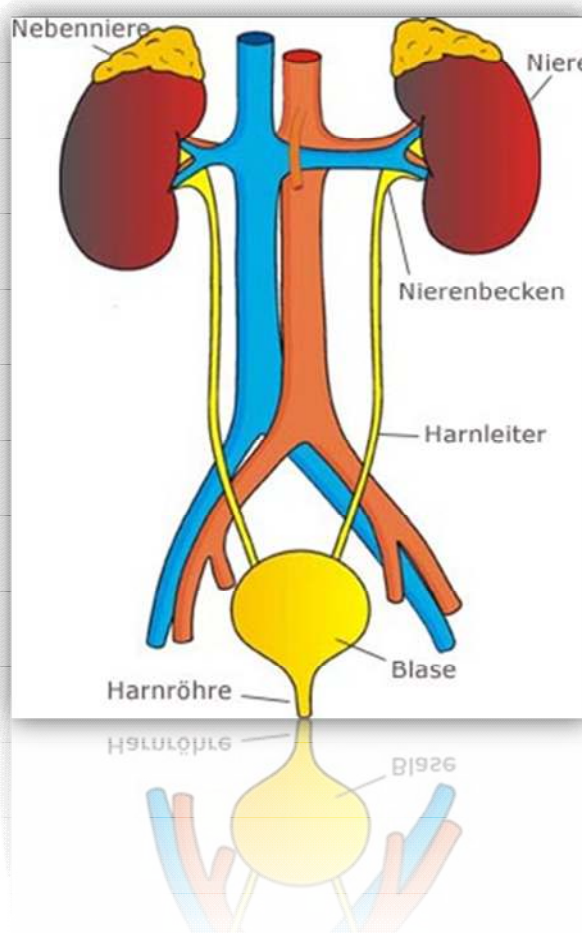
Die sogenannte RPI-Theorie besagt, dass Menschen mit einem höheren BMI (wobei wir in unserem Modell keinen größeren Wert auf das Verhältnis zwischen Muskel- und Fettmasse gelegt haben) länger mit Dialyse überleben. Wir nehmen daher an, dass die Toxine nicht nur aus der Interzellularflüssigkeit und dem Blut, sei es durch die Nieren oder durch das Dialyseverfahren, gefiltert, sondern auch in Fett und Muskeln gespeichert oder abgebaut werden. Weiters ist wichtig, dass die Ärzte wesentlich genauer bestimmen können, wie oft bzw. wie stark die Dialyse ausgeführt werden muss.



Die Niere

Die Funktion der Nieren besteht unter anderem darin Giftstoffe (Toxine) aus dem Blut zu filtern und durch Bildung des Primärharns zu binden. Davon werden täglich ca. 150l produziert. Würde man alles davon ausscheiden, müsste man mehrere Liter Wasser pro Minute trinken! Man verlöre auch einen riesigen Prozentsatz an Mineralien. Daher stellt die Niere ein lebenswichtiges Organ dar; ohne sie könnte man nur 2 bis 3 Tage überleben. Deswegen werden in der Niere wiederum viele Inhaltsstoffe zurück in das Blut befördert. Nur Toxine, die unter anderem als Nebenprodukte in Verdauungsorganen entstehen, werden, neben Wasser und überflüssigen Mineralien, in der Harnblase gesammelt und dann ausgeschieden.

Bei Versagen oder Verlust gibt es leider nur zwei erfolgsversprechende Behandlungsformen, die ein Überleben der betroffenen Probanden zu einem befriedigend hohen Prozentsatz ermöglichen: Einerseits gibt es die Möglichkeit einer Transplantation, die jedoch von seltenen Spenderorganen abhängig und daher nicht immer möglich ist. Die Hämodialyse liefert daher eine willkommene Alternative, die im Gegensatz zur Transplantation mit einem geringeren Risiko verbunden ist. Sie ermöglicht eine relativ „normale“ Lebensführung, stellt aber gleichzeitig durch die ständige Behandlung eine extreme Belastung für den Körper dar, weshalb die durchschnittliche Lebensdauer nach Versagen der Niere nur etwa 10 Jahre beträgt.

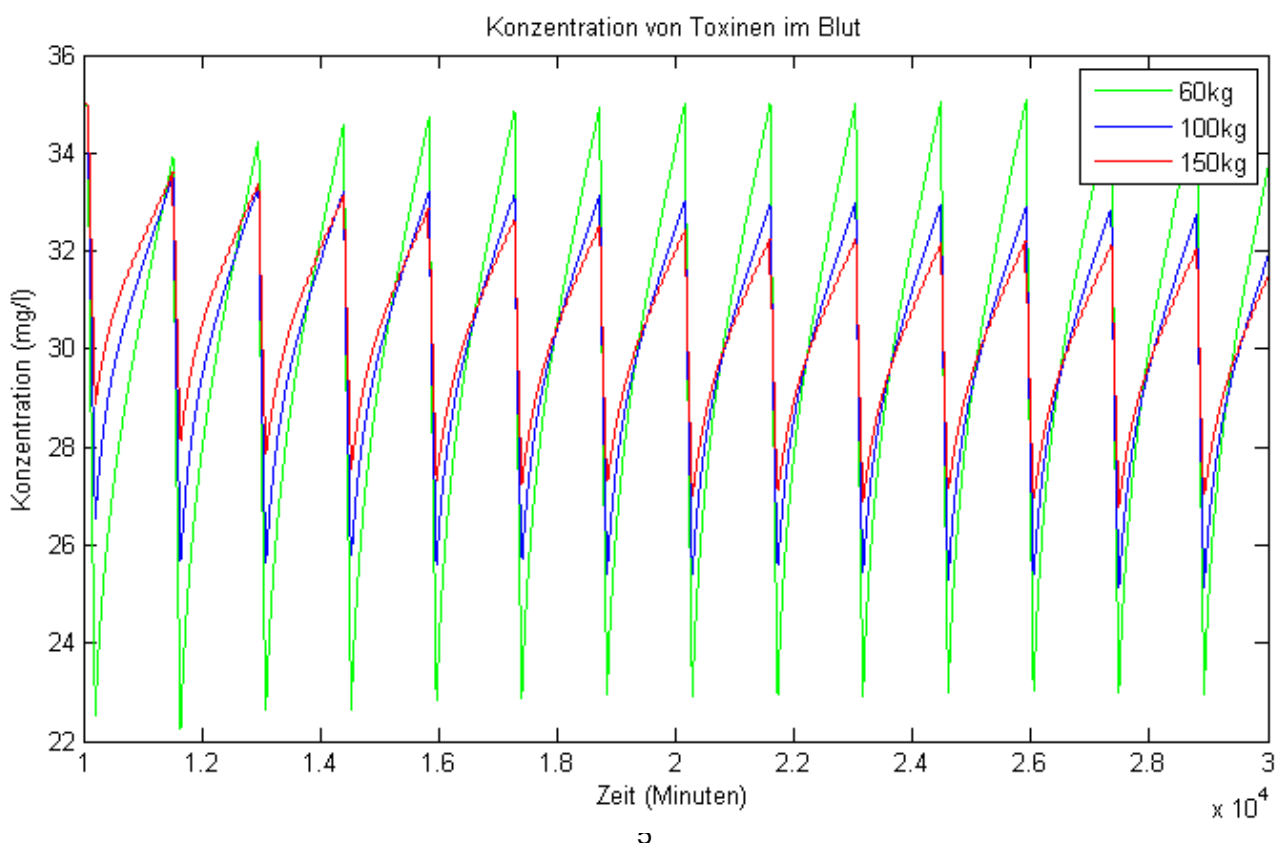


Modellierung der Dialysebehandlung

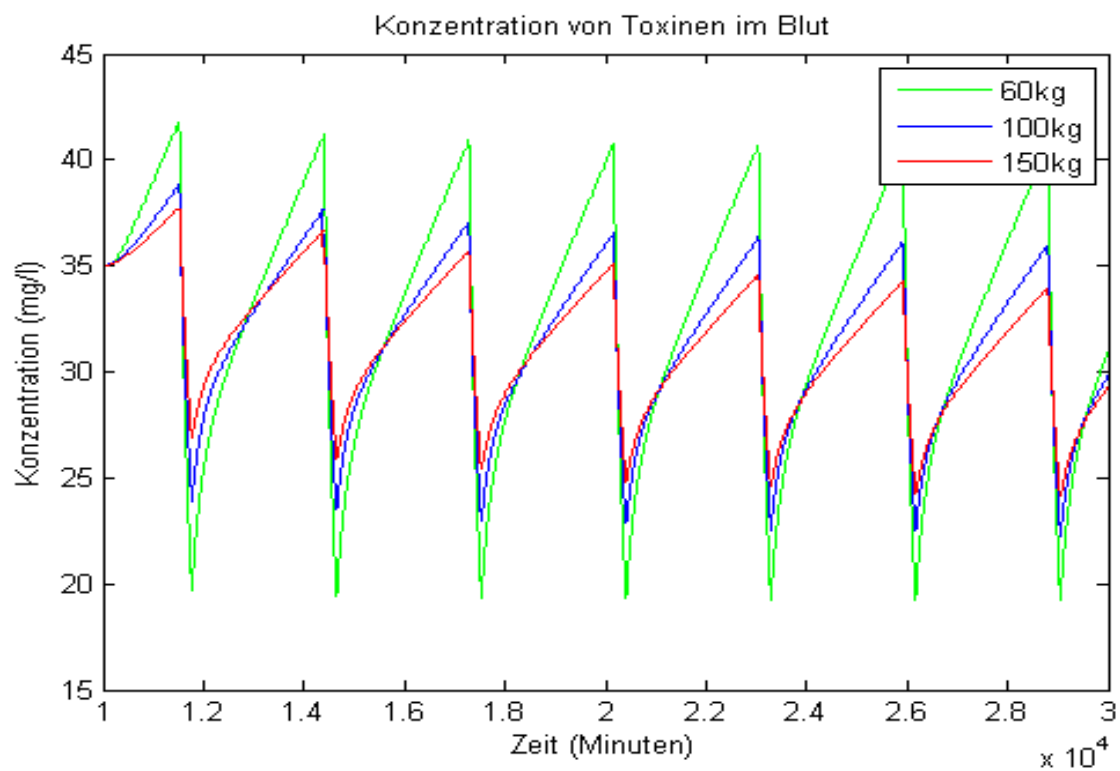
Der Erfolg einer Dialysebehandlung ist maßgeblich von der Häufigkeit der Dialysesitzungen abhängig, wobei es diese abhängig von Körpergewicht, Nierenrestfunktion, Herzleistung individuell abzustimmen gilt. Meist ist es sinnvoll, die Dialysebehandlung häufig, dafür kürzer durchzuführen, da die Dialyseeffektivität in der ersten Behandlungshälfte am höchsten ist. Weiters wird durch häufige Dialyse der Durchschnittswert der Toxinkonzentration im Blut gesenkt. In der Regel wird eine Dialyse 4-5 Stunden mind. 3mal in der Woche durchgeführt. Unter anderem gibt es auch die Möglichkeit einer Heimdialyse, die kürzere Dialyseintervalle und eine gleichmäßigere Therapie ermöglicht. Mit der täglichen Hämodialyse werden Vorteile der Peritonealdialyse (täglich) und der konventionellen Hämodialyse (hohe Effektivität) miteinander kombiniert. Für Patienten, die diese Möglichkeit nicht haben, ist die Nachtdialyse, bei welcher viele Patienten nach einiger Zeit schlafen können, eine angenehme Alternative um die Dialysezeit zu verlängern.

Die drei angeführten Graphen beschreiben die Toxinkonzentration im Blut (mg/l) von einer je 1,8m großen und 60kg, 80kg bzw. 150 kg schweren Person. Beim ersten Graphen erfolgt die Dialyse nach einem Tag zu je 2 Stunden, beim zweiten nach 2 Tagen zu je 4 Stunden und beim dritten nach 3 Tage zu je 5 Stunden. Wie man anhand dieser Graphen eindeutig erkennen kann, ist der durchschnittliche Toxingehalt im Blut bei einer täglichen Dialyse weitaus geringer als bei einer Dialyse, die alle 3 Tage durchgeführt wird. Weiters wird der Vorteil von Patienten mit geringerem BMI besonderes im dritten Graphen deutlich. Die Schwankungen der Toxinkonzentration im Blut und damit auch die Belastung auf den menschlichen Körper sind bedeutend höher.

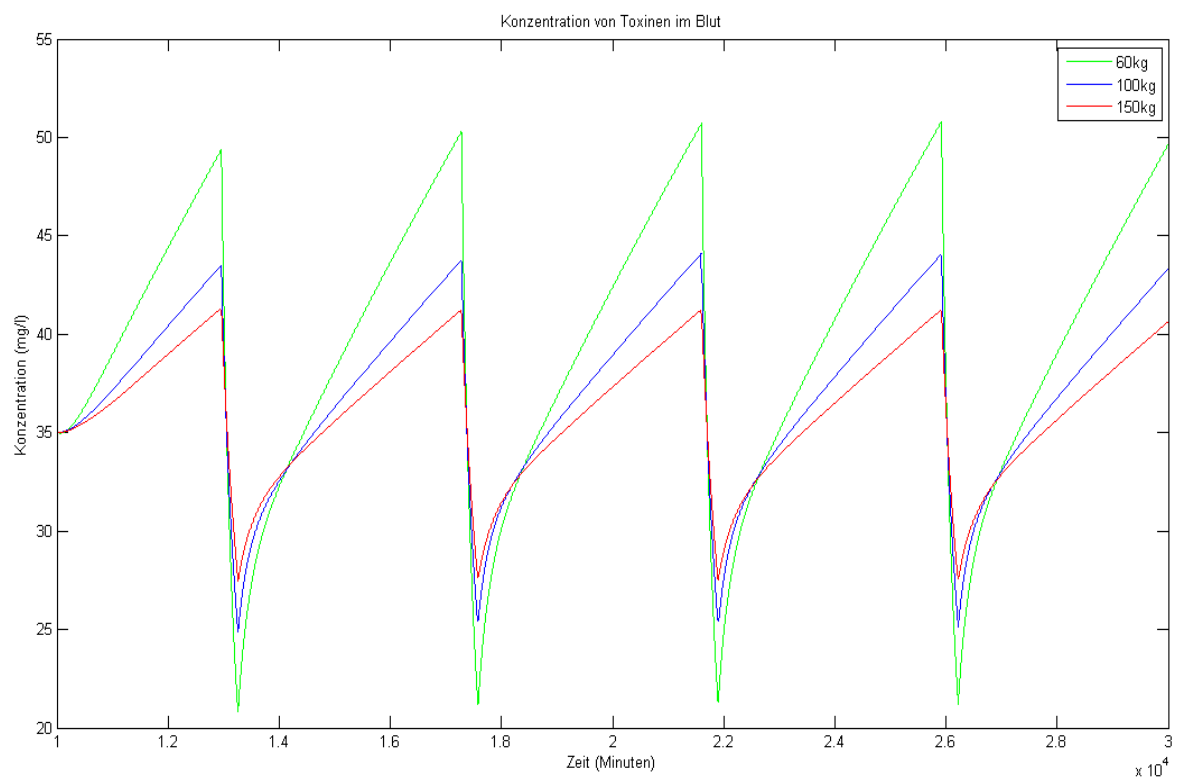
Tägliche Dialyse (1440 min):



Dialyse nach 2 Tagen (2880 min):



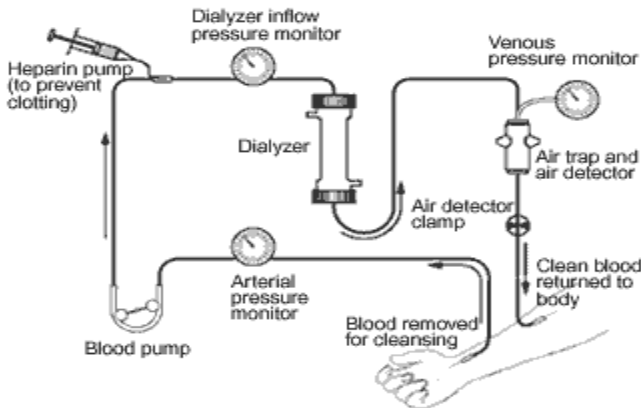
Dialyse nach 3 Tagen (4320 min):



Die Hämodialyse

Die Hämodialyse beschreibt ein maschinelles Blutreinigungsverfahren, welches die Funktion der Niere weitgehend ersetzt. Sie funktioniert mit der Annahme, dass fast alle Stoffwechselgifte im Plasma bzw. Wasseranteil des Blutes gelöst sind; vereinfacht ausgedrückt entfernt die Maschine mit

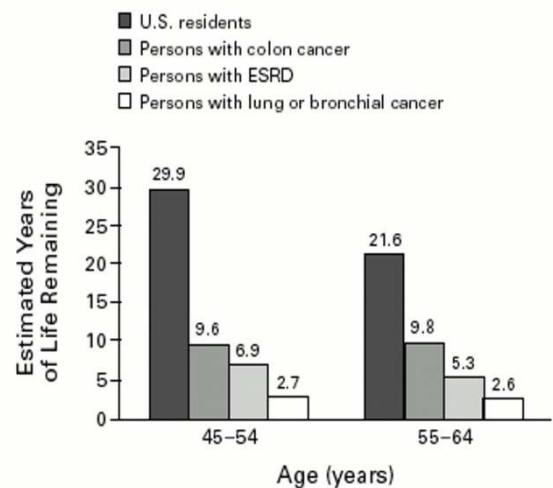
verschiedenen Chemikalien einen Teil des Wassers und somit auch einen Großteil des Toxins aus dem Blut.



Der Toxin- und Plasmaspiegel im Blut, welcher ohne die Regulation der Niere kontinuierlich und lebensgefährlich ansteigt, wird durch die Dialyse bestenfalls auf ein normal hohes Level gesenkt. Dieser drastische Anstieg bzw. Abnahme des Toxin- und Plasmagehalts im Organismus stellt eine enorme Belastung dar und ist der

Grund für die kurze Lebenserwartung von Dialysepatienten. Dabei lässt sich feststellen, dass eine häufige, dafür kürzer durchgeführte Dialyse effektiver ist. Der Durchschnitt des Toxinspiegels im Organismus ist bei einer intensiveren Behandlung wesentlich niedriger und kleinere Schwankungen dieses Wertes stellen eine geringere Belastung für den menschlichen Körper dar. Während in Amerika die Behandlung nur zweimal für je fünf Stunden in der Woche durchgeführt wird, führt man den Dialyseprozess dreimal in der Woche zu 4 Stunden durch.

Verglichen mit anderen Krankheiten, wie rechts in der Grafik dargestellt, wird ersichtlich, dass zum Beispiel Personen mit Lungenkrebs eine wesentlich kürzere Lebenserwartung aufweisen als Menschen, die an einer Niereninsuffizienz leiden und mit Dialyse behandelt werden; andererseits leben Menschen mit anderen Arten von Krebs auch deutlich länger, je nachdem, ob der Krebs nun lebenswichtige Körperregionen betrifft oder nicht.



ESRD: Niereninsuffizienz in der Endstufe
Pastan Sand Bailey J. N Engl J Med 1998;338:1428-1437

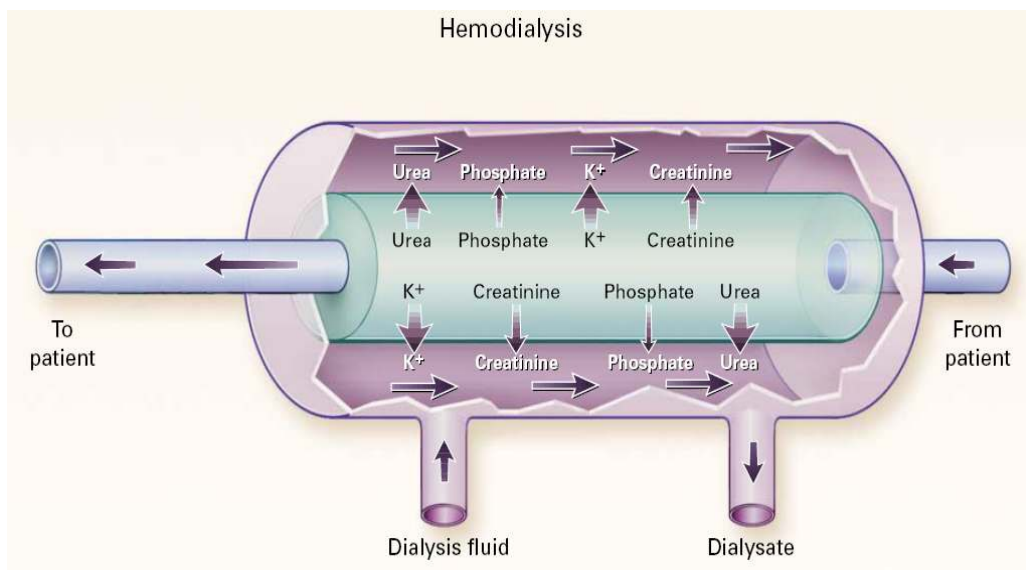
Annahmen für die effektiven Volumen

- $V_B = 0,38 * \text{Körpergewicht}$
 - effektives Volumen für Wasser im Körper, exklusive Wasser in Zellen
- $V_F = 0,189 * \text{BMI}$
 - effektives Fettvolumen
 - wenn Gewicht kleiner 50kg, dann $V_F = 0.034 * \text{Körpergewicht}$
- $V_M = \{1,5\text{l} - 60\text{kg}, 3\text{l} - 100\text{kg}, 6,5\text{l} - 150\text{kg}\}$
 - effektives Muskelvolumen

Sonstige Konstanten

- $k_M(t) = 0,03177$ (Austausch mit Muskelgewebe)
- $k_F(t) = 0,0193$ (Austausch mit Fettgewebe)
- $k_D(t) = 0,010$ (Giftabnahme bei der Dialyse pro Minute)
- $k_T(t) = 0.3$ (Giftzunahme pro Minute)
- $k_{MA}(t) = k_{FA}(t) = 0,001$ (Abbaukoeffizient in Muskeln und Fett)

Dialysis Bild von Forni L and Hilton P. N Engl JMed 1997;336:1303-1309



Kompartimentenmodell

Mithilfe eines Kompartimentenmodells kann man durch Vereinfachung diverser Einheiten bestimmte Werte wie etwa Massen- oder Konzentrationswerte bestimmen. Hierfür teilt man beispielsweise den Körper in mehrere, imaginäre Einheiten. (In der Regel sind drei Einheiten für die weitergehenden Berechnungen besonders hilfreich)

Eine Einheit wird dann durch zwei (oder auch mehr) Parameter charakterisiert. In der beigefügten Grafik sieht man den konstanten Faktor K_{01} , der die Konzentration des Kompartiments mit dem Volumen V vergrößert. Der zweite Faktor k_{10} hingegen sorgt dafür, dass die durch k_{01} stetig vergrößerte Konzentration sinkt.



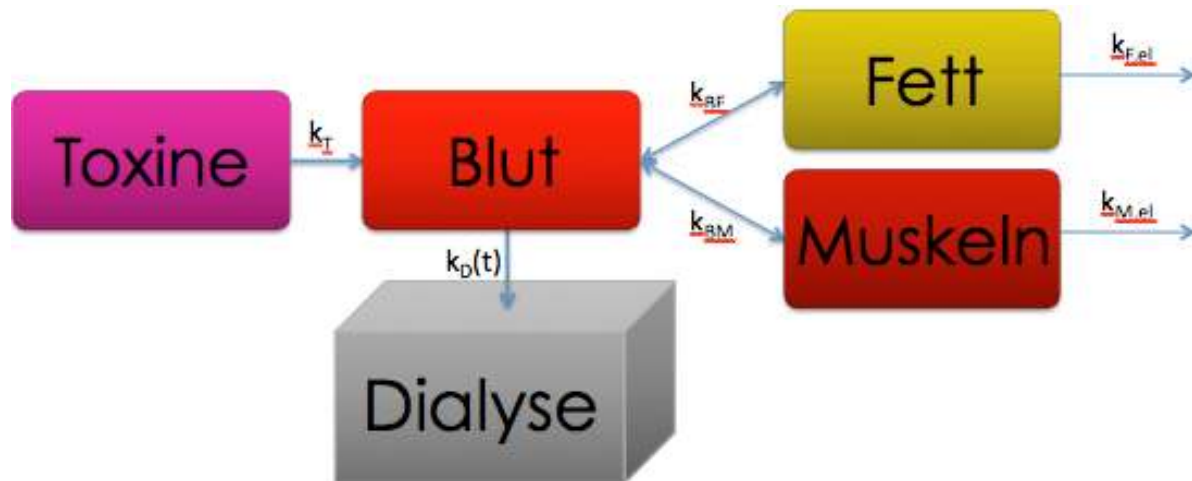
In unserem Fall waren dies das angegebene Volumen und die jeweiligen Austauschkonstanten. Anhand dieser Parameter lassen sich in weiterer Folge Dinge wie Masse oder Konzentration berechnen. In der Regel ist der Austausch der Massenwerte proportional zur Konzentration der Massen in den Kompartimenten.

Die Aufstellung eines solchen Kompartimentenmodells findet besonders im Bereich der Medizin und Pharmazie großen Anklang, da man mit diesem Giftgehalte beziehungsweise Konzentrationen aller Art relativ simpel berechnen kann.

Kompartimentanalyse für Hämodialyse

Auch wir haben uns mit einer Problemstellung aus diesem Bereich beschäftigt: Wir behandelten den Prozess der Toxinkonzentration im menschlichen Körper, der normalerweise durch die Nieren konstant reguliert wird. In unserem Modell beschrieben wir aber den Toxingehalt bei Hämodialysepatienten, der aufgrund der nur periodisch durchführbaren Dialyse bis zum Zeitpunkt der „Blutwäsche“ stark ansteigt, dann aber durch die Dialyse rapide absinkt. Dies stellt, wie bereits in Kapitel 2 dargelegt, eine herbe Belastung für den Körper dar, weshalb bei Patienten mit nicht voll funktionstüchtigen beziehungsweise ohne Nieren statistisch gesehen eine höhere Mortalitätsrate zu verzeichnen ist. Des Weiteren ist zu beobachten, dass Dialysepatienten mit einem höheren BMI (Body Mass Index, $= m/h^2$) im Gegensatz zu Patienten mit einem niedrigeren BMI offenbar eine größere Überlebenschance besitzen (siehe Kapitel über FMS). Wir gehen daher davon aus, dass jene sich im Blut beziehungsweise in der Interzellularflüssigkeit befindenden Toxine zusätzlich sowohl in Fett- als auch in Muskelgewebe gespeichert werden. Hierbei sei angemerkt, dass jene Toxine in der Interzellularflüssigkeit bei unserem Modell zu jenen im Blut dazugerechnet werden. Aus diesem Grund beinhaltet das „Blut-Kompartiment“ in diesem Bericht sowohl die Interzellulartoxine als auch die sich tatsächlich im Blut befindenden Gifte.

Demnach teilen wir in unserem Modell den menschlichen Körper in drei Kompartimente, um den Teilbereich der Entgiftung möglichst genau widerzuspiegeln: Blut, Fett- und Muskelgewebe. Zusätzlich definieren wir Austauschkonstanten, die eine fixe Geschwindigkeit für die Verteilung im Körper vorgeben und Konstanten für Erzeugung, Abbau und Filterung (Dialyse) von Giften.



Bestimmte Konstanten (in diesem Fall k_{BF} und k_{BM}) bewegen sich zwischen zwei Kompartimenten (Blut-Fett; Blut-Muskeln) auf Grund des chemischen Prozesses der Diffusion jedoch in beide Richtungen gleichermaßen.

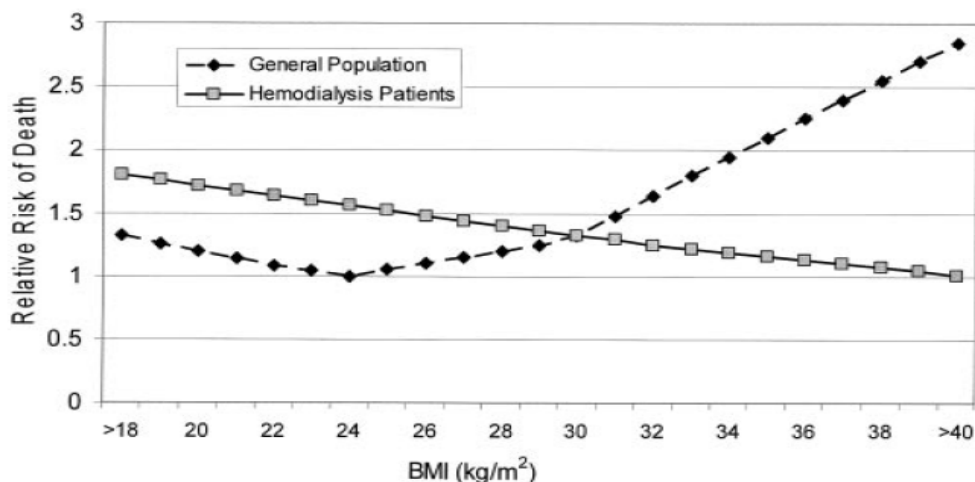
Mithilfe dieser Annahmen lässt sich ein Gleichungssystem erstellen, mit dem man vorerst die jeweiligen Massen der Kompartimente und später deren Konzentrationen berechnen kann.

FMS-Theorie

(FAT-MEN-SURVIVING)

Durch verschiedene durchgeführte Studien hat sich ergeben, dass Menschen mit einem höheren BMI (30+) durchschnittlich länger leben als dünnere Menschen, wenn sie auf eine Dialyse angewiesen sind. Zwar steigt die Lebensdauer mit einem höheren BMI, aber auf der anderen Seite sinkt diese mit einem niedrigeren Gewicht pro Quadratmeter.

Veranschaulichen soll dies folgendes Diagramm, welches die durchschnittliche Bevölkerung (schwarz) und Dialysepatienten (grau) mit dem jeweiligen Body Mass Index zeigt:

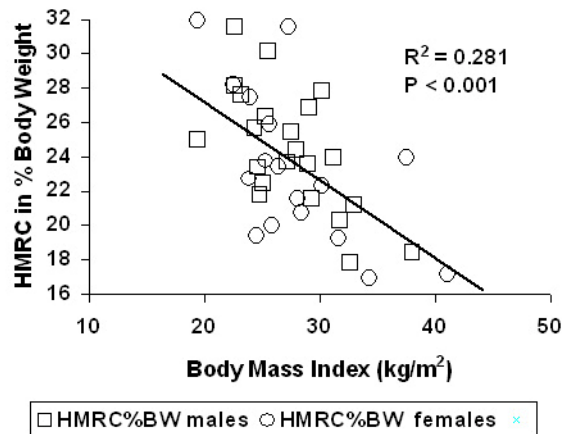


Kalantar-Zadeh, 2006

Wie man sieht, erreicht man normal mit einem Body Mass Index um 24 durchschnittlich eine optimale Lebensdauer, welche bei einem höheren BMI nicht gegeben ist. Zu hohe oder zu niedrige Werte implizieren gesundheitliche Probleme und bringen damit ein kürzeres Leben mit sich. Anders bei Dialysepatienten, welche durch die graue Linie sichtlich dargestellt werden und somit das Fundament unserer Hypothese bilden. Deutlich zu erkennen ist hierbei die höhere Lebensdauer bei einem höheren Wert des Body Mass Index, welches ein wichtiger Anhaltspunkt unserer FMS-Theorie ist.

Grundsätzlich besagt diese Theorie nun, dass Dialysepatienten mit einem BMI länger leben, wenn dieser Wert größer ist. Beruhend auf Statistiken und Studien können wir mathematische Modelle für den Beweis dieser Vermutung aufstellen. Auf diesen Daten basierend war ebenso eine Zusammenstellung von einigen Annahmen möglich:

- Adipöse Dialysepatienten leben durchschnittlich länger
- Fett- und Muskelmasse spielen eine wichtige Rolle beim Speichern und Abbauen der hämatischen Toxine
- Stoffwechselaktivität verringert sich relativ zum Gewicht bei steigendem BMI

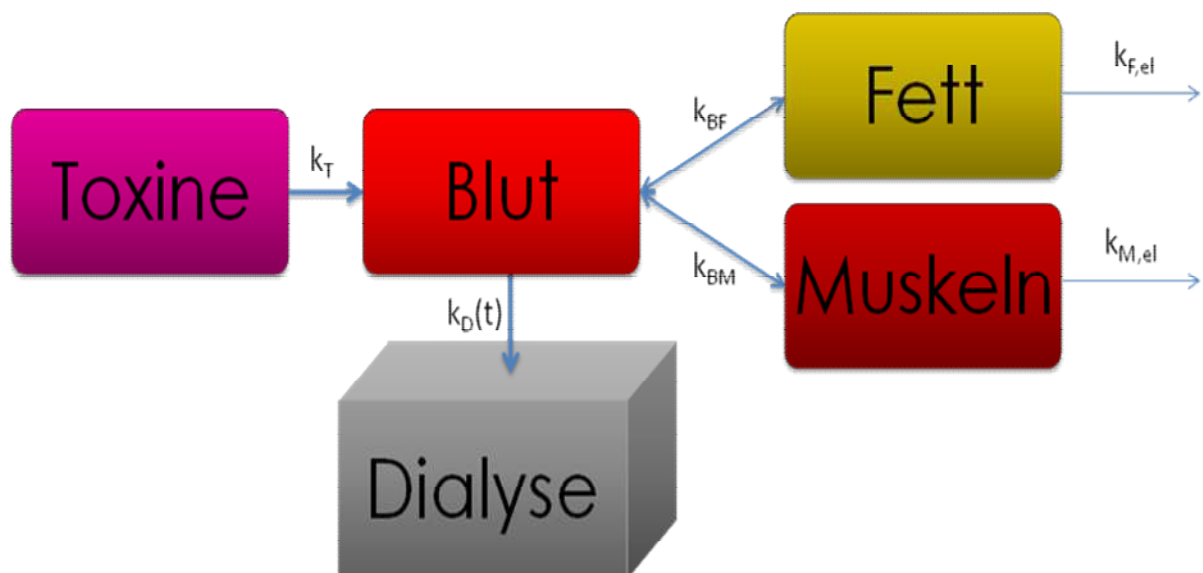


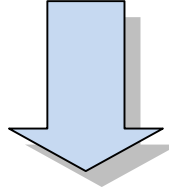
Stoffwechselaktivität Sarkar, Kidney Int 2006

Nachdem vage Hypothesen einen Beweisvorgang durchleben müssen, bedarf auch unsere Annahme eines solchen Vorgangs unterzogen zu werden. Für die Bestätigung dieser wird ein Differenzialgleichungssystem aufgestellt, um ein graphisches Darstellen zu ermöglichen. Dieses System beinhaltet repräsentative Werte für die absolute Toxinmasse und die Konzentration der toxischen Substanzen in Blut, Muskel- und Fettgewebe.

Beweis

Wir haben nun, basierend auf dem Kompartimentenmodell, ein Gleichungssystem, welches die Dialyse eines Patienten mit beliebigen Körpermaßen darstellen kann, wobei jedem Kompartiment ein Volumen, eine Masse und jedem möglichen Austauschweg ein Faktor zugeordnet wird, von denen einer, der Dialysefaktor, zeitabhängig ist, aufgestellt. Der Dialysefaktor wird mit einer modulo-Funktion kontrolliert, die ihn alle 2 Tage (2880 Minuten) auf den angegebenen Faktor setzt und sonst bei 0 belässt.





$$\frac{dM_B}{dt} = k_T + k_{FB} \frac{M_F}{V_F} + k_{MB} \frac{M_M}{V_M} - k_{BF} \frac{M_B}{V_B} - k_{BM} \frac{M_B}{V_B} - k_D(t) \frac{M_B}{V_B}$$

$$\frac{dM_F}{dt} = k_{BF} \frac{M_B}{V_B} - k_{F,el} \frac{M_F}{V_F} - k_{FB} \frac{M_F}{V_F}$$

$$\frac{dM_M}{dt} = k_{BM} \frac{M_B}{V_B} - k_{M,el} \frac{M_M}{V_M} - k_{MB} \frac{M_M}{V_M}$$

Die linke Seite des Systems gibt nun für eine beliebigen Zeitintervall die Änderung der

Toxinmenge in Blut/extrazellulärer Flüssigkeit ($\frac{dM_B}{dt}$), Muskelmasse ($\frac{dM_M}{dt}$) und Fettgewebe ($\frac{dM_F}{dt}$) an. Dabei ist die Angabe der richtigen Koeffizienten und ihrer Werte von großer Bedeutung:

k_T	= Toxinproduktion	= 0.3 mg/ min
k_{FB}	= Fett-Blut-Diffusion	= 0,0193
k_{BF}	= Blut-Fett-Diffusion	= 0,0193
k_{MB}	= Muskel-Fett-Diffusion	= 0,03177
k_{BM}	= Blut-Muskel-Diffusion	= 0,03177
$k_{F,el}$	= Abbaurrate im Fett	= 0,001
$k_{M,el}$	= Abbaurrate im Muskel	= 0,001
$k_D(t)$	= Dialyserate	= 0,10

Für die Gleichung sind auch die Effektivwerte der verschiedenen Kompartimentvolumen nötig, die die Realwerte der gesamten Massen (z.B. Fett im ganzen Körper) teilweise nicht erreichen. Diese kann man mit einfachen Formeln berechnen, welche aus medizinischen Studien stammen.

$$V_B = 0.038 \times \text{BW l}$$

$$V_F = 0.0189 \times \text{BMI} - 2.92 \text{ l}$$

$$V_M = 6.5\text{l (schwer)} \quad 3\text{l (mittel)} \quad 1.5\text{l (leicht)}$$

Um das Gleichungssystem nun in MATLAB verwenden zu können, muss es in eine Matrix umgewandelt werden, welche MATLAB dann in kleinen Intervallen auswerten und als Graf Dracula darstellen kann. Man spaltet das System in eine Matrix A und die Spaltenvektoren M, $\dot{M}$ und B auf. A enthält alle Volumen und Koeffizienten die mit Variablen verbunden sind, B die Konstante k_T , welche den Toxinanstieg angibt, M alle Toxinmassen und $\dot{M}$ gibt schließlich die Veränderung der Toxinmassen pro Minute an.

$$\dot{M} = A \cdot M + B$$

$$A = \begin{pmatrix} \frac{-k_{BF} - k_{BM} - k_D(t)}{V_B} & \frac{k_{FB}}{V_F} & \frac{k_{MB}}{V_M} \\ \frac{k_{BF}}{V_B} & \frac{-k_{F,el} - k_{FB}}{V_F} & \mathbf{0} \\ \frac{k_{BM}}{V_R} & \mathbf{0} & \frac{-k_{M,el} - k_{MB}}{V_M} \end{pmatrix}$$

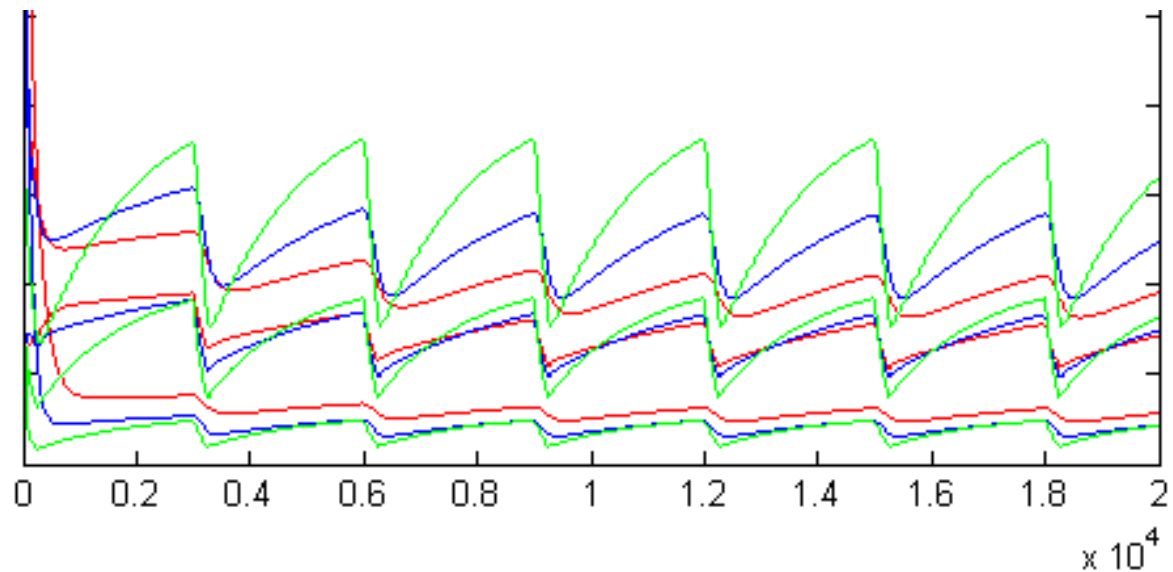
$$\dot{M} = \begin{pmatrix} \frac{dM_B}{dt} \\ \frac{dM_F}{dt} \\ \frac{dM_M}{dt} \end{pmatrix} \quad M = \begin{pmatrix} M_B \\ M_F \\ M_M \end{pmatrix} \quad B = \begin{pmatrix} k_T \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}$$

Um nun wirklich unsere gewünschten Graphen zu erhalten benötigen wir noch die Gewichte der untersuchten Personen, welche bei dem Berechnungsbefehl in MATLAB angegeben werden.

Das Programm produziert die Toxinmassenwerte für die drei Kompartimente, welche nur mehr in Konzentrationen umgewandelt werden müssen. Diese lässt man als Diagramm ausgeben, um Übersicht über die Auswirkungen der Dialyse auf die diversen Körperteile zu erhalten.

Der Graph sieht nun im Normalzustand folgendermaßen aus:

Farben: 150kg = rot
 100kg = blau
 60kg = grün

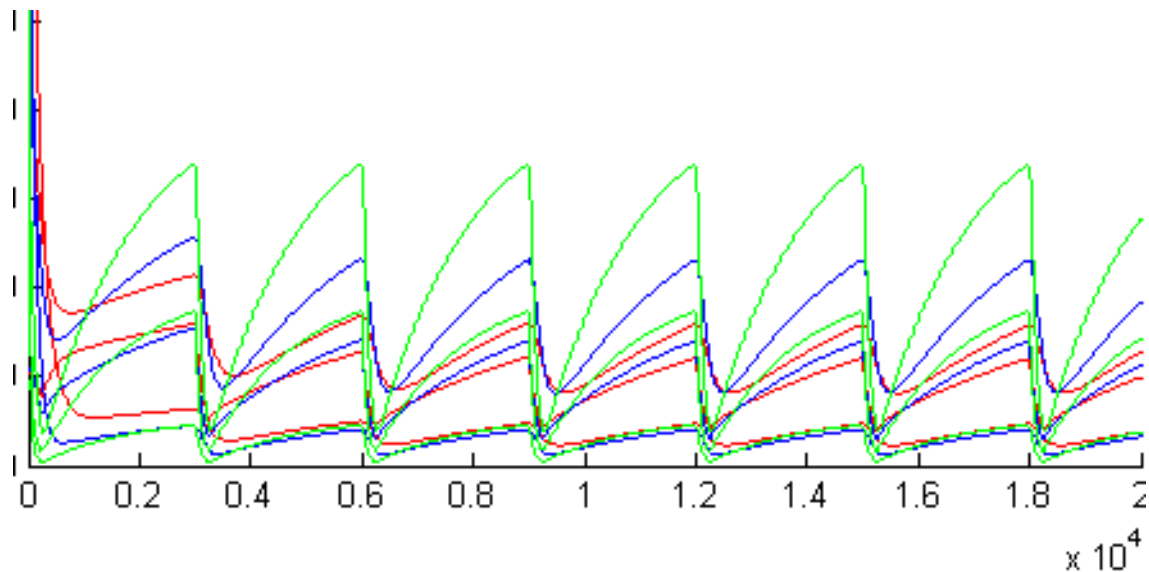


In diesem Diagramm sind die Fettwerte die niedrigsten, die Blutwerte die mittleren und die Muskelwerte die höchsten. Man muss, wenn man eine Analyse durchführen will den rechten Bereich betrachten, da sich die Werte erst einpendeln müssen.

Die Blutwerte zeigen die ungleichmäßigsten Verläufe, da sie direkt von der Dialyse betroffen sind. Die Fettwerte zeigen dagegen nur sehr träge Veränderungen, da sie einen geringeren Austauschfaktor zum Blut haben.

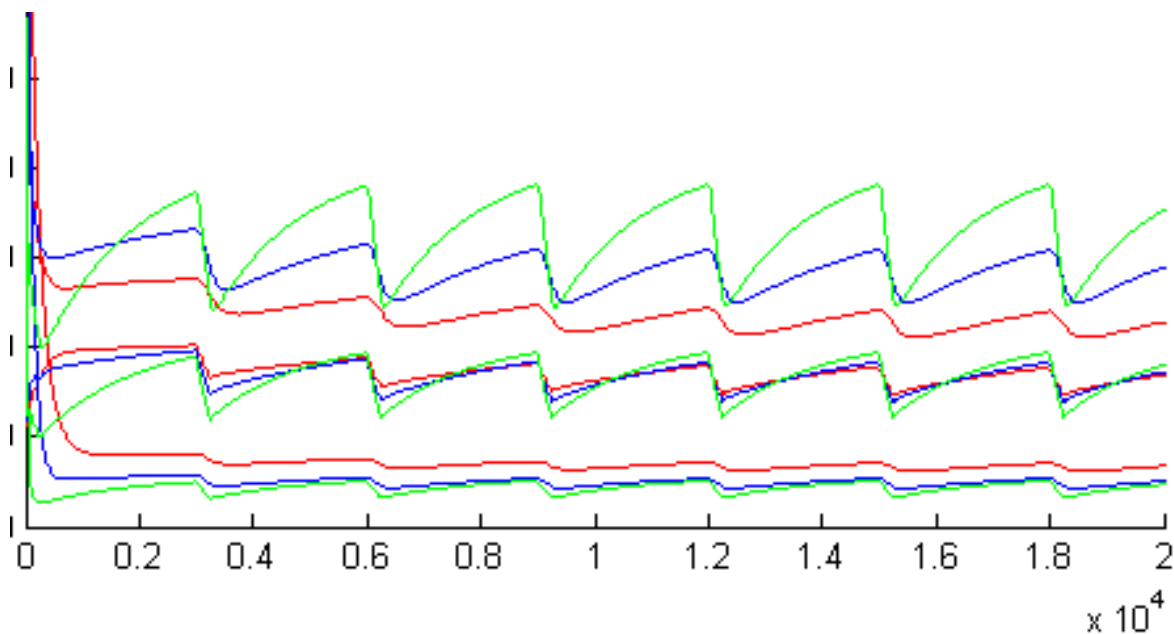
Der am Diagramm zwar schwer zu erkennen aber durch eine Durchschnittsberechnung leicht zu ermittelnde Mittelwert der 150kg-Person ist niedriger als der, der 100kg- und 60kg-Person. Man kann nun zum Beispiel den Wert von $k_D(t)$, der die Dialyse kontrolliert, vergrößern oder verkleinern wobei interessante Folgen am Diagramm zu erkennen sind.

Höhere Dialyserate: $k_D(t)=0,3$



Man kann hier nun erkennen, dass der durchschnittliche Toxinlevel des Blutes bei der 150kg-Person am stärksten gesunken ist, jedoch mit Abstand am gleichmäßigsten ist.

Niedrigere Dialyse rate: $k_D(t)=0.05$

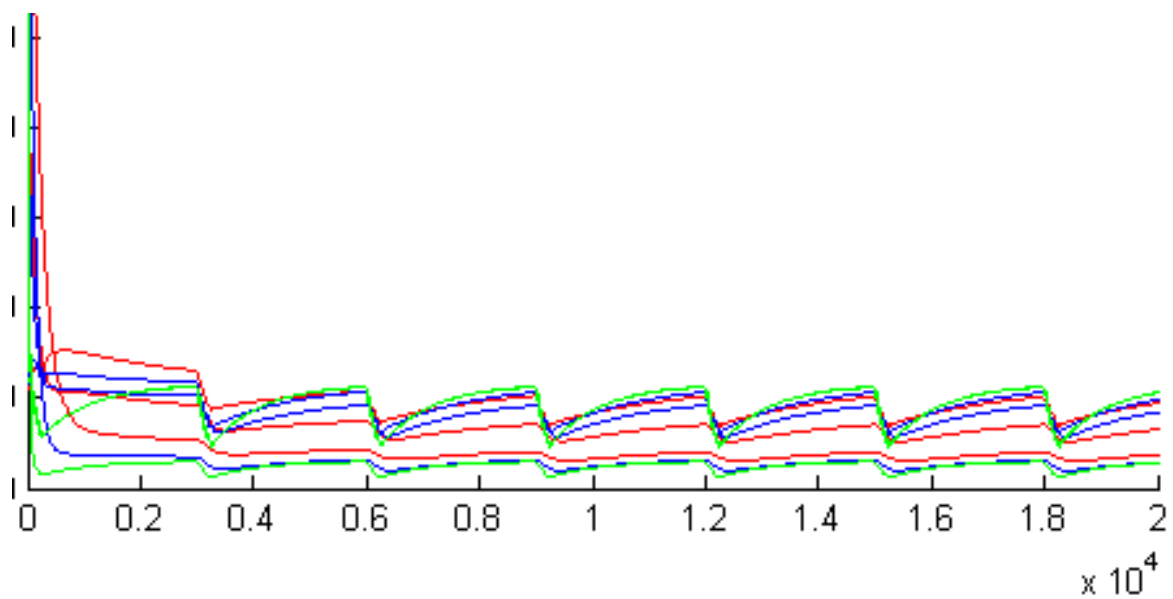


Wenn man im Gegenversuch den Dialysefaktor senkt erhält man auch das umgekehrte Ergebnis. Nun ist der Wert der 150kg-Person gleich dem oder höher als jener der leichteren Personen. Daraus folgt, dass vor allem bei hohen Dialysewerten, daher auch in Zukunft mit neueren Dialysegeräten, dickere Menschen eher einen Vorteil haben.

Damit ist die FSM-Theorie bestätigt, soweit man unsere Berechnungsdaten als richtig, sinnvoll und aktuell festlegen kann.

Man kann allerdings auch den Abbaufaktor in den Muskeln oder im Fett verändern. So werden alle Kurven um einiges spitzer wenn man annimmt, dass keine Toxine abgebaut werden, oder um einiges runder, wenn man den Faktor höher einstellt, wobei auch alle Konzentrationen näher zusammenrücken.

Höherer Abbau im Muskel: $k_{M,el}=0,05$



Fragen für die Zukunft

- Ausgleichen von Messfehlern
- Genauere Messung der Effektivwerte und Parameter
- Genauere Unterscheidung zwischen effektiven Fett- und Muskelvolumen
- Erforschung der Sinnhaftigkeit von absichtlicher Gewichtszunahme von Dialysepatienten

Matlab

Es ist derzeit eine der wichtigsten technisch/mathematischen, von der Firma MathWorks.Inc entwickelten, Programmiersprachen. Sie verwendet fast alle gängigen Programmiermethoden und besitzt auch sehr viele vorgefertigte Funktionen.

Der Semikolon ; wird am Ende einer Zeile verwendet, um z.B. Ausgaben von Variablen zu unterdrücken; verpflichtend ist er nicht.

Die wichtigsten Befehle/ Funktionen:

- Variablen
- Matrizen
- Schleifen
 - *for-Schleife*
 - *while-Schleife*
- Abfrage
 - *if-Abfrage*
- Zeichenfunktionen
 - *plot*
 - *surf*

Variablen

Variablen brauchen in Matlab nicht deklariert werden, ihr Typ wird automatisch bestimmt. Man muss sie in den seltensten Fällen umwandeln.

Beispiel:

```
a=10;           %die Variable a hat jetzt den Wert 10
```

Matrizen

Matrizen (Sg.:Matrix) sind ein oder mehrdimensionale Datenfelder (auch Arrys in anderen Programmiersprachen)

Die Schleife

Als Schleife wird im Allgemeinen ein Prozess bezeichnet, der sich beliebig oft wiederholen lässt.

Die for-Schleife:

Bei der for-Schleife wird eine Variable bei jedem Durchlauf erhöht. Sie endet erst wenn die Variable den, im Vorhinein festgelegten, Wert erreicht. Das Ende des zu wiederholenden Bereiches wird durch das Wort „end“ festgelegt.

Beispiel:

```
for i=1:10      %Der Anfang wird festgelegt, i ist die Schleifenvariable, läuft 10 mal  
Befehle        %Hier werden die zu wiederholenden Befehle angegeben  
end            %Hier wird der zu wiederholende Bereich beendet
```

Die while-Schleife:

Die while-Schleife führt einen gewissen Bereich (wieder durch end begrenzt) immer wieder durch, solange die Bedingung erreicht wird (das Erhöhen einer Variablen ist nicht zwingend nötig).

Beispiel:

```
while i<10      %Initialisierung der while-Schleife, Bedingung (läuft solange i<10)  
Befehle        %Zu wiederholende Befehle
```


end % Ende des zu wiederholenden Bereichs

Die Abfrage

Durch eine Abfrage kann eine einfache oder mehrfache Verzweigung erzielt werden, dies ist nötig um dynamisch auf geänderte Variablenwerte reagieren zu können.

Die if-Abfrage:

Die if-Abfrage ist die einfachste Art der Abfrage. Sie besteht eigentlich nur aus einer Bedingung, den auszuführenden Befehlen, wenn die Bedingung erfüllt ist und abermals „end“ um den auszuführenden Bereich abzugrenzen.

Beispiel:

```
if i<10                      % Beginn der Abfrage, wird ausgeführt, wenn die Variable i<10 ist
Befehle                    % Befehle die ausgeführt werden, wenn die Bedingung erfüllt ist
end                         % Begrenzung der Befehle die auszuführen sind
```

Zeichenfunktionen

- plot: zur Darstellung von eindimensionalen Matrizen
- surf: zur Darstellung von dreidimensionalen Matrizen

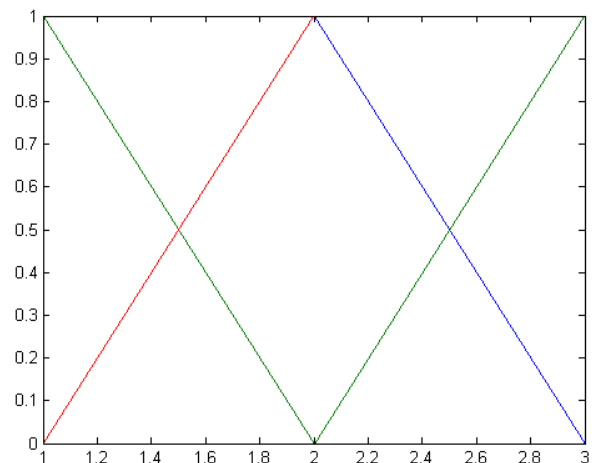
Handhabung von Bildern

Als erstes ist es wichtig die Intensitäten des Bildes auszulesen. In Matlab sind alle Intensitäten einer zweidimensionalen Matrix gespeichert, entweder Bitweise (Werte zwischen 0 und 255) oder in Werten zwischen 0 und 1.

Ein Beispiel:

```
1 1 0
A = 1 0 1
0 1 1
```

Das Bild wäre 3 Pixel (Bildpunkte) lang und 3 Pixel hoch. 0 bedeutet Schwarz und 1 bedeutet Weiß. Es wären also 3 schwarze Pixel, die die Linie bilden, in einem Feld von weißen Pixeln. (Der Strich würde normalerweise nur aus drei Pixeln bestehen, aber für eine bessere Darstellung hat er in diesem Beispiel mehr.)



1) Hydrostatik

Schwimmlagen von Balken mit quadratischem Querschnitt

Aufgabenstellung:

Unser Ziel ist es, herauszufinden welche Stellungen des Körpers einer theoretischen Schwimm Lage entsprechen. Damit ein Schwimmzustand vorherrscht muss die Wasserlinie mit der Verbindungsline von Körper - und Verdrängungsschwerpunkt einen rechten Winkel bilden. (das Skalare Produkt der Vektoren muss 0 sein). Zusätzlich lassen sich noch 3 verschiedene Schwimm lagen unterscheiden:

1) Stabile Schwimm lage:

- Die potentielle Energie der Eintauch lage erreicht hier ein lokales Minimum
- Ist die Schwimm lage, die wir auch in der Natur immer finden
- Bei kleinen Störungen wie Wellen oder Wind fällt der Körper immer wieder in diese Lage zurück

2) Instabile Schwimm lage:

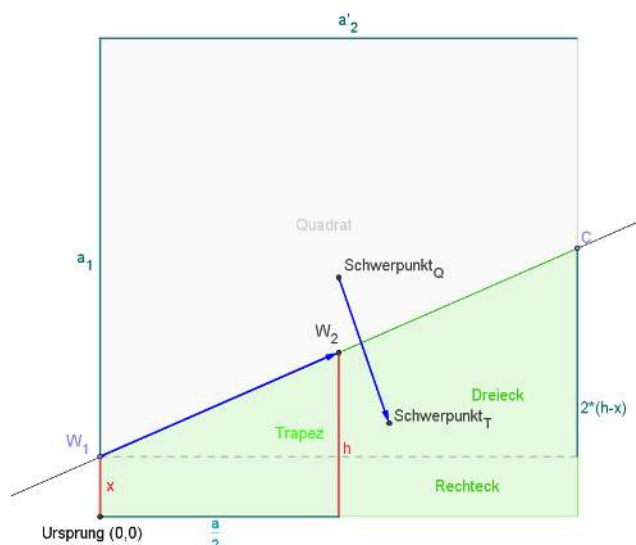
- Die potentielle Energie der Eintauch lage erreicht hier ein lokales Maximum
- Diese Schwimm lage ist nur theoretisch und kommt in der Natur nicht vor
- Schon durch kleine Störungen fällt der Körper in eine andere Lage

3) Indifferente Schwimm lage:

- Die potentielle Energie der Eintauch lage ist in jeder Position die selbe
- Wie beispielsweise bei einem kreisförmigen Querschnitt

Eine Grundlage für unsere Überlegungen war das **Archimedische Prinzip**. Es besagt, dass ein Körper dann schwimmt, wenn das Gewicht des Körpers gleich groß ist, wie das Gewicht der von ihm verdrängten Flüssigkeit.

Hier der Querschnitt von dem die weiteren Überlegungen ausgehen mit den passenden Beschriftungen:



1) Trapezschwimmlagen

1.1) Geometrische Vorbereitungen:

Flächenberechnungen:

$$\begin{aligned} F_D &= a * (h - x) \\ F_R &= a * x \\ F_T &= a * h \end{aligned}$$

F_D ... Fläche Dreieck
 F_R ... Fläche Rechteck
 F_T ... Fläche Trapez

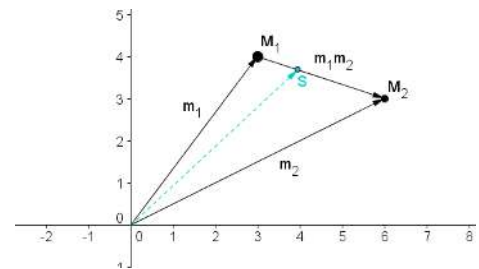
Schwerpunktberechnungen:

$$\begin{aligned} \vec{S_Q} &= \left(\frac{a}{2}, \frac{a}{2} \right) \\ \vec{S_D} &= \left(\frac{2a}{3}, \frac{2h+x}{3} \right) \\ \vec{S_R} &= \left(\frac{a}{2}, \frac{x}{2} \right) \end{aligned}$$

S_Q ... Schwerpunkt
 Quadrat
 S_D ... Schwerpunkt Dreieck
 S_R ... Fläche Rechteck

1.2) Weg zum Trapezschwerpunkt (ST):

Anfangs standen wir vor dem Problem, dass wir noch keine Formel für die Berechnung des Trapezschwerpunktes hatten und so teilten wir es in Rechteck und Dreieck. Wenn man nun die beiden einzelnen Schwerpunkte berechnet, kann man auf den Trapezschwerpunkt schließen. Dieser muss auf der Verbindungsstrecke der einzelnen Schwerpunkte liegen. Wo genau der Punkt liegt, ergibt sich durch das Verhältnis der Massen (in unserem Fall der Flächen). Der gemeinsame Schwerpunkt liegt näher bei der schwereren Masse bzw. der größeren Fläche.



$$\begin{aligned} \vec{s} &= \vec{m_1} + \frac{M_2}{M_1 + M_2} * (\vec{m_2} - \vec{m_1}) \\ \vec{s} &= \frac{M_1}{M_1 + M_2} * \vec{m_1} + \frac{M_2}{M_1 + M_2} * \vec{m_2} \end{aligned}$$

Umgelegt auf unser Beispiel setzen wir jetzt anstatt der Massen die Fläche ein. Dadurch ergibt sich folgende Formel:

$$\vec{S_T} = \frac{F_R}{F_T} * \vec{S_R} + \frac{F_D}{F_T} * \vec{S_D}.$$

Jetzt setzen wir die vorher ausgerechneten Flächen- und Schwerpunktsformeln ein und erhalten die Koordinaten des Trapezschwerpunktes.

$$\begin{aligned} &= \frac{a * x}{a * h} * \left(\frac{a}{2} \middle| \frac{x}{2} \right) + \frac{a * (h - x)}{a * h} * \left(\frac{2a}{3} \middle| \frac{2h + x}{3} \right) \\ &= \left(\frac{ax}{2h} \middle| \frac{x^2}{2h} \right) + \left(\frac{2ah - 2ax}{3h} \middle| \frac{(2h + x) * (h - x)}{3h} \right) \\ &S_T = \left(\frac{a(4h - x)}{6h} \middle| \frac{4h^2 - 2hx + x^2}{6h} \right) \end{aligned}$$

1.3) Berechnung der Vektoren:

Vektor zwischen den beiden Schwerpunkten:

$$\begin{aligned} \overrightarrow{S_Q S_T} &= \overrightarrow{S_T} - \overrightarrow{S_Q} \\ &= \left(\frac{a(4h-x)}{6h} \middle| \frac{x^2 + 4h - 2hx}{6h} \right) - \left(\frac{a}{2} \middle| \frac{a}{2} \right) \\ \overrightarrow{S_Q S_T} &= \left(\frac{a(h-x)}{6h} \middle| \frac{x^2 + 4h^2 - 2hx - 3ah}{6h} \right) \end{aligned}$$

Richtungsvektor der Wasserlinie:

$$\begin{aligned}\overrightarrow{W_1 W_2} &= \overrightarrow{W_2} - \overrightarrow{W_1} \\ &= \left(\frac{a}{2} \middle| h\right) - (0 \middle| x) \\ \overrightarrow{W_1 W_2} &= \left(\frac{a}{2} \middle| h - x\right)\end{aligned}$$

1.4) Skalarprodukt

Aufstellen des Skalaren Produkt zur Überprüfung der Lage zwischen den Vektoren (Bei 0 stehen sie normal).

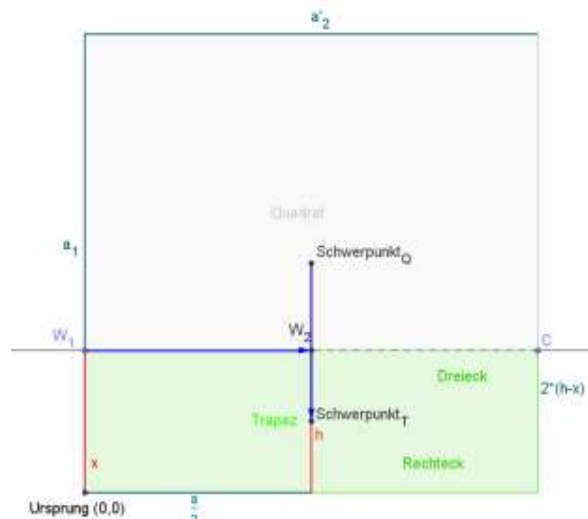
$$\begin{aligned} \overrightarrow{W_1 W_2} * \overrightarrow{S_Q S_T} &= \left(\frac{a}{2} \middle| h - x \right) * \left(\frac{a(h-x)}{6h} \middle| \frac{x^2 + 4h^2 - 2hx - 3ah}{6h} \right) \\ &= \frac{a^2(h-x)}{12h} + \frac{x^2h + 4h^3 - 2h^2x - 3ah^2 - x^3 - 4h^2x + 2hx^2 + 3ahx}{6h} \quad | * 2 \\ \overrightarrow{W_1 W_2} * \overrightarrow{S_Q S_T} &= \frac{a^2h - a^2x + 6hx^2 + 8h^3 - 12h^2x - 6ah^2 - 2x^3 + 6ahx}{12h} \end{aligned}$$

Wir stellen nun die Frage für welche x -Werte dieses skalare Produkt null wird. Das heißt wir suchen die Nullstellen des Polynoms:

$$-2x^3 + 6hx^2 - a^2x - 12h^2x + 6ahx + a^2h + 8h^3 - 6ah^2.$$

Wir wissen bereits dass der Körper immer dann schwimmt, wenn $x = h$ ist (der Körper Unterwasser also ein Rechteck ist). Somit ergibt sich der erste Linearfaktor den wir für die Polynomdivision verwenden können.

$$\begin{aligned} & (-2x^3 + 6hx^2 + x * [-a^2 - 12h^2 + 6ah] + a^2h + 8h^3 - 6ah^2): (x - h) \\ & = -2x^2 + 4xh - a^2 - 8h^2 + 6ah \end{aligned}$$



Im nächsten Schritt setzen wir dieses Polynom null und berechnen die Nullstellen.

$$A = -2, B = 4h, C = -a^2 - 8h^2 + 6ah$$

$$x_{1,2} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

$$x_{1,2} = \frac{-4h \pm \sqrt{-48h^2 - 8a^2 + 48ah}}{-4}$$

Unter diesen Lösungen sind nur jene x-Werte interessant, für die gilt: $0 < x \leq h$.

Die Nullstellen sind nur dann reell (komplexe Nullstellen kommen für unser Problem nicht in Frage), wenn die Diskriminante nicht negativ ist. Wir untersuchen jetzt für welche h-Werte die Diskriminante nicht negativ ist, das heißt:

$$-48h^2 - 8a^2 + 48ah \geq 0$$

$$h^2 + \frac{a^2}{6} - ah \leq 0$$

$$\left(h - \frac{a}{2}\right)^2 - \frac{a^2}{4} + \frac{a^2}{6} \leq 0$$

$$\left(\frac{a}{2} - h\right)^2 \leq \frac{a^2}{12}$$

Weil beide Zahlen unter dem Quadrat positiv sind, können wir beidseits die Wurzel ziehen.

$$h \geq \frac{a}{2} * \left(1 - \frac{1}{\sqrt{3}}\right)$$

Wenn also $h < \frac{a}{2} * \left(1 - \frac{1}{\sqrt{3}}\right)$ ist, dann gibt es keine reellen Lösungen der Quadratischen Gleichung und daher auch keine echten Trapez-Schwimmlagen. Wenn aber $h \geq \frac{a}{2} * \left(1 - \frac{1}{\sqrt{3}}\right)$ ist, dann müssen wir untersuchen, ob die zugehörigen Lösungen der quadratischen Gleichung zwischen 0 und h liegen.

Es gibt 3 Fälle:

- a) Beide Lösungen liegen außerhalb des Bereiches, das heißt es gibt keine Trapezschwimmlagen
- b) Genau eine Lösung liegt im Bereich, dann gibt es genau eine Trapezschwimmlage
- c) 2 verschiedene Lösungen liegen im Bereich, dann gibt es 2 Trapezschwimmlagen (diesen Fall haben wir geometrisch in unseren Konstruktionen jedoch nie vorgefunden)

1.5) Stabilitätsuntersuchung

Die potentielle Energie (E_{pot}) zu jeder Eintauchsituation ist proportional zum Höhenunterschied von Körper- und Verdrängungsschwerpunkt. Errechnet wird nun die Funktion für die potentielle Energie aus dem skalaren Produkt zwischen dem normalen Einheitsvektor der Wasserlinie und dem Verbindungsvektor zwischen Körper- und Verdrängungsschwerpunkt ($\vec{v}$).

$$\vec{n} = \frac{\vec{w}^\perp}{|\vec{w}|} = \frac{(2(x-h)|a)}{\sqrt{4(h-x)^2 + a^2}}$$

$$\vec{v} = \frac{1}{6h} (a(x-h)|-x^2 + 2hx + 3ah - 4h^2)$$

$$E_{\text{pot}} = \vec{n} * \vec{v}$$

$$= \frac{1}{6h * \sqrt{4(h-x)^2 + a^2}} (2a(x-h)^2 + a(-x^2 + 2hx + 3ah - 4h^2))$$

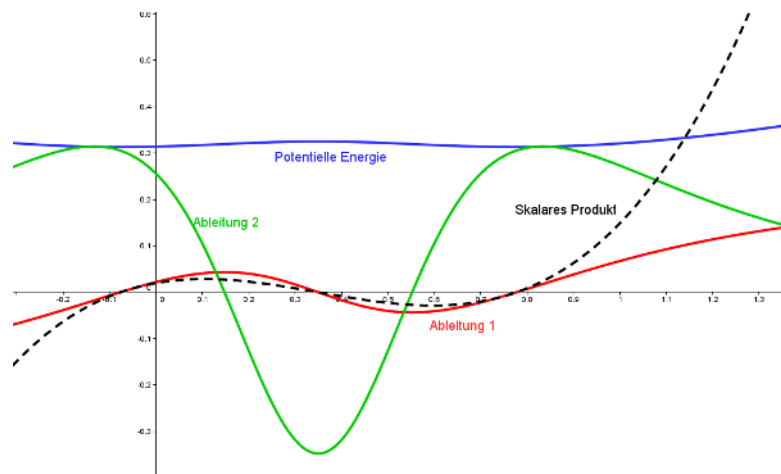
Zu vollständigem Quadrat ergänzt:

$$= \frac{1}{6h * \sqrt{4(h-x)^2 + a^2}} (2a(x-h)^2 + a(-(x-h)^2 + 3ah - 3h^2))$$

$$E_{\text{pot}}(x) = \frac{a}{6h} * \frac{(x-h)^2 + 3ah - 3h^2}{\sqrt{4(h-x)^2 + a^2}}$$

Durch diese Funktion und deren Ableitungen können wir auf die stabilen Schwimmlagen schließen:

- ➔ Wo $E_{\text{pot}}(x)' = 0$ (hat die gleichen Nullstellen wie das skalare Produkt) haben wir überhaupt Schwimmlagen.
- ➔ Zu diesen x-Werten brauchen wir die 2. Ableitung die uns dann Auskunft über die stabilen Schwimmlagen gibt:
 1. Wenn sie in einem x-Wert positiv ist, heißt das, dass dort die potentielle Energie ein Minimum hat ➔ eine stabile Schwimmlage liegt vor
 2. Ein negativer x-Wert liefert uns eine instabile Schwimmlage



2) Dreiecksschwimmlagen

2.1) Geometrische Vorbereitungen

Flächenberechnung:

$$F_D = \frac{(a - y) * z}{2}$$

$$z = \frac{2ah}{a - y}$$

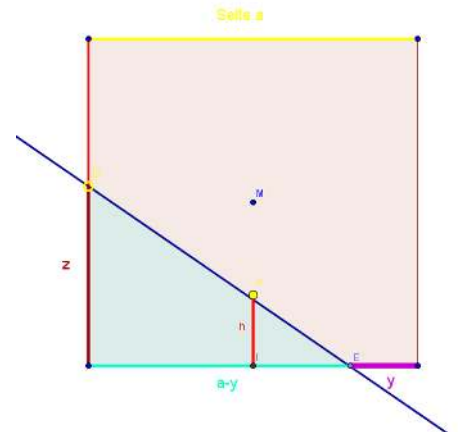
Schwerpunktsberechnung:

$$S_D = \left(\frac{2a - 2y}{3} + y \mid \frac{2ah}{3a - 3y} \right)$$

Vektorberechnungen:

$$\overrightarrow{W_1 W_2} = \left(a - y \mid \frac{2ah}{a - y} \right)$$

$$\overrightarrow{S_Q S_D} = \left(\frac{a + 2y}{6} \mid \frac{4ah - 3a^2 + 3y}{6a - 6y} \right)$$



2.2) Berechnung des skalaren Produkts:

$$\overrightarrow{W_1 W_2} * \overrightarrow{S_Q S_D} = \left(a - y \mid \frac{2ah}{a - y} \right) * \left(\frac{a + 2y}{6} \mid \frac{4ah - 3a^2 + 3y}{6a - 6y} \right) =$$

$$\left(\frac{a + 2y}{6} \right) * (a - y) + \left(\frac{4ah - 3a^2 + 3y}{6a - 6y} \right) * \left(\frac{2ah}{a - y} \right) =$$

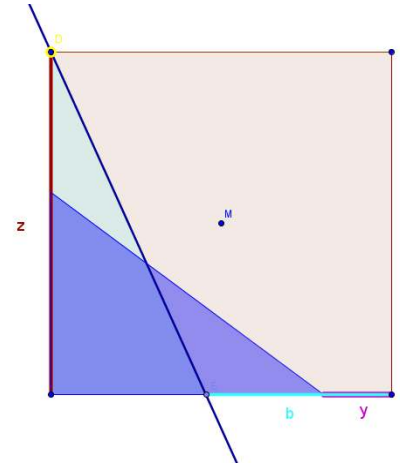
$$\frac{a^2 + 2ay - ay - 2y^2}{6} + \frac{8a^2h^2 - 6a^3h + 6a^2yh}{6a^2 - 6ay - 6ay + 6y^2} =$$

$$\frac{a^2 + ay - 2y^2}{6} + \frac{8a^2h^2 - 6a^3h + 6a^2yh}{6 * (a^2 - 2ay + y^2)} =$$

$$\frac{(a^2 + ay - 2y^2) * (a^2 - 2ay + y^2) + 8a^2h^2 - 6a^3h + 6a^2yh}{6 * (a^2 - 2ay + y^2)} =$$

$$\frac{a^4 - a^3y - 3a^2y^2 + 5ay^3 - 2y^4 + 8a^2h^2 - 6a^3h + 6a^2yh}{6 * (a^2 - 2ay + y^2)}$$

Dann werden über das Nullsetzen des Zählers wieder die Nullstellen berechnet. Diese liefern uns die y-Werte, für welche unter Wasser eine Dreiecksschwimmlage entsteht. Von diesen Nullstellen sind für unser Problem aber nur jene relevant, welche im Intervall $[0, b]$ liegen. „b“ steht für den maximalen y-Wert, welcher dafür sorgt, dass das Dreieck noch innerhalb unseres quadratischen Querschnitts liegt. Herleitung für b:



Grenzfall:

$$z = \frac{2ah}{a-y} \quad z = a; \quad y = b$$

$$a^2 - ab = 2ah$$

$$b = \frac{a^2 - 2ah}{a}$$

$$b = a - 2h$$

2.3) Stabilitätsüberprüfung:

$$\vec{w} = \left(a - y \left| \frac{2ah}{a-y} \right. \right) \quad \vec{w}^\perp = \left(-\frac{2ah}{a-y} \left| a - y \right. \right) \quad |\vec{w}| = \sqrt{(a-y)^2 + \left(\frac{2ah}{a-y} \right)^2}$$

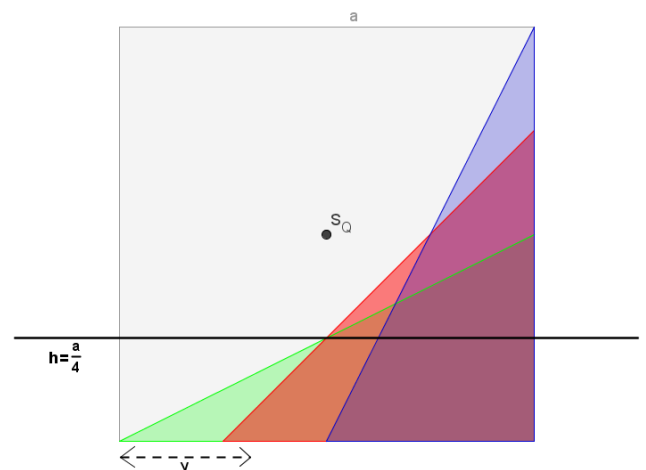
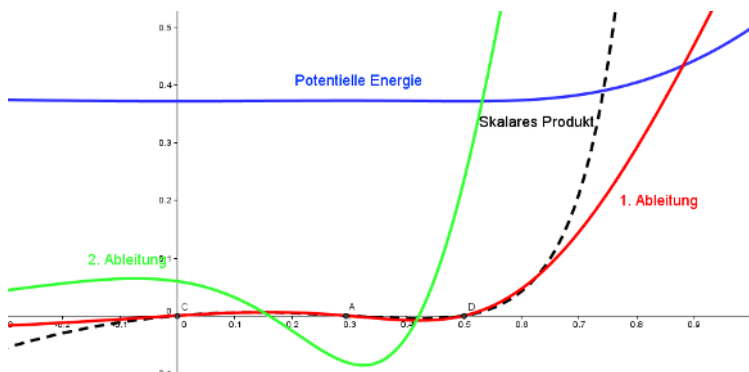
$$\vec{n} = \frac{\vec{w}^\perp}{|\vec{w}|} = \frac{1}{\sqrt{\frac{a^4 - 4a^3y + 4a^2h^2 + 6a^2y^2 - 4ay^3 + y^4}{(a-y)^2}}} * \left(-\frac{2ah}{a-y} \left| a - y \right. \right)$$

$$\vec{v} = \left(\frac{-a - 3y}{6} \left| \frac{-4ah + 3a^2 - 3ay}{6 * (a-y)} \right. \right)$$

Hier gilt das Gleiche wie für den Trapezfall. Man benötigt nur diese Formel!

$$E_{Pot} = \vec{n} * \vec{v} = \frac{1}{\sqrt{\frac{a^4 - 4a^3y + 4a^2h^2 + 6a^2y^2 - 4ay^3 + y^4}{(a-y)^2}}} * \left(-\frac{2ah}{a-y} \left| a - y \right. \right) * \left(\frac{-a - 2y}{6} \left| \frac{-4ah + 3a^2 - 3ay}{6(a-y)} \right. \right)$$

$$E_{Pot} = \frac{3a^3 - 2a^2h - 6a^2y + 8ahy + 3ay^2}{6 * (a-y) * \sqrt{\frac{a^4 - 4a^3y + 4a^2h^2 + 6a^2y^2 - 4ay^3 + y^4}{(a-y)^2}}}$$



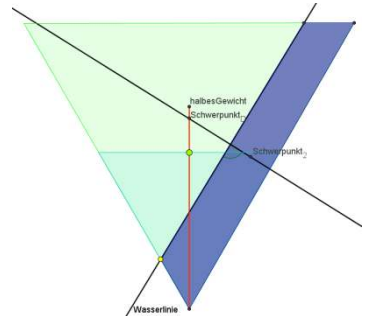
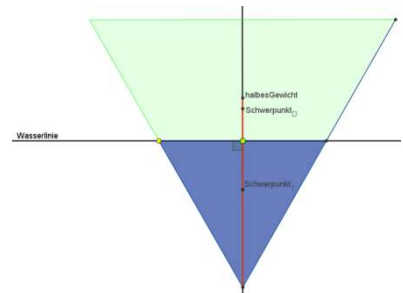
3) Balken mit einem Querschnitt von einem gleichseitigen Dreieck

Dieser Fall wurde nur geometrisch untersucht, da der rechnerische Vorgang sehr komplex ist und viel Zeit in Anspruch nimmt.

Der Querschnitt wird in zwei flächengleiche Teile geteilt, da sich sonst die ganzen Schwimmlagen wiederholen.

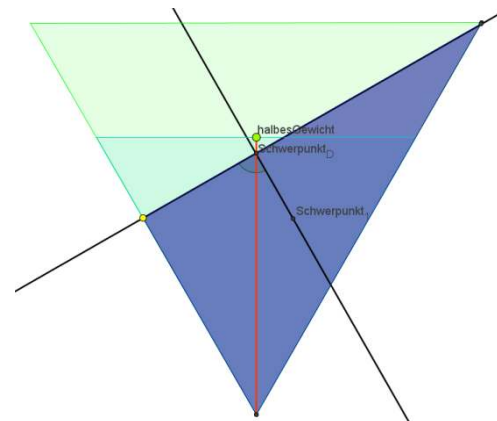
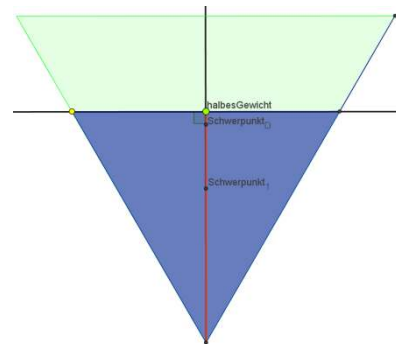
Hierbei gibt es jetzt wieder 2 Fälle zu unterscheiden. Wenn das spezifische Gewicht (bzw. die Fläche) des unterwasserliegenden Teil des Querschnitts kleiner ist, als die Hälfte des spezifischen Gewichts des ganzen Querschnitts, so ergeben sich folgende Schwimmlagen:

- Eine Dreiecksschwimmlage, wo die Wasserlinie parallel zu einer Seite des Querschnitts ist.
- Eine Vierecksschwimmlage, wenn die Wasserlinie ebenfalls parallel zu einer Seite des Querschnitts ist.



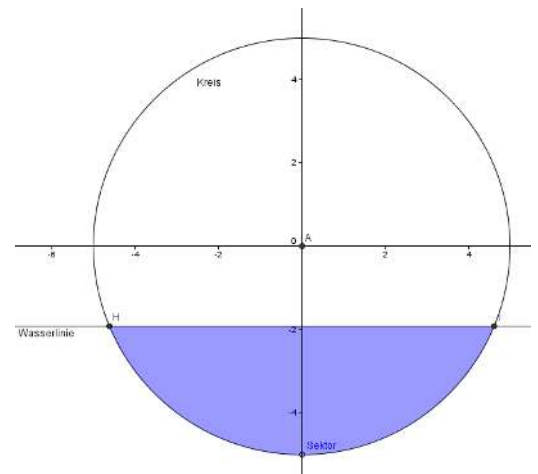
Wenn das spezifische Gewicht des unter Wasser liegenden Körpers gleich dem halben spezifischen Gewicht des gesamten Querschnitts ist, dann ergeben sich folgende Schwimmlagen.

- Eine Dreiecksschwimmlage, wenn die Wasserlinie parallel zu einer Seite des Querschnitts ist.
- Eine Dreiecksschwimmlage, wenn die Wasserlinie die Höhe auf eine Seite ist (normal zur Seite + Durchgang durch die gegenüberliegende Höhe)



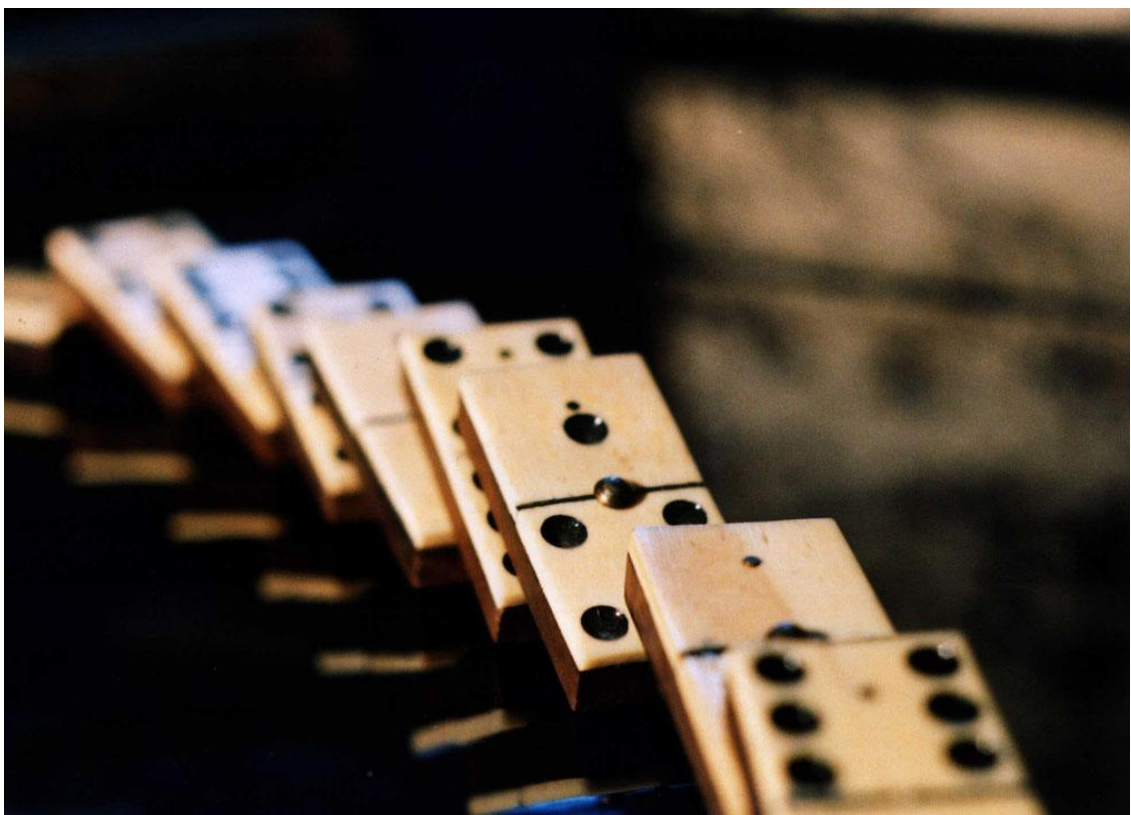
4) Kreisquerschnitt:

Einen Balken mit kreisförmigem Querschnitt haben wir nur konstruiert, nicht berechnet. Hier gibt es nur die indifferente Schwimmelage, da ein Balken mit einem solchen Querschnitt in jeder Lage schwimmt, jedoch durch die kleinste Einwirkung einer Kraft in eine andere Lage gebracht werden kann, die wiederum indifferent ist.



MODELLIERUNGSWOCHE 2010

vom 07.02.2010 bis 13.02.2010 in Pöllau



Der Domino-Effekt

Ein Projekt aus der Mechanik mit Prof. Bernd THALLER

erarbeitet von:

Theresa ABL ■ Eva DITTMER ■ Anna FALLMANN ■ Sebastian GÖLLY ■ Alexander PFLEGER ■ Jakob STEINBAUER

1. Einführung

1.1. Einleitung

1.2. Problemstellung

2. Mathematische Annäherung an das Problem

2.1. Geometrie

2.2. Dynamische Größen

2.3. Energiebetrachtung

2.4. Steigung

2.5. Exponentielles Wachstum

1.Einführung

1.1. Einleitung

4,8 Millionen, das war die Anzahl der Steine die am Freitag, 13.11.2009 zu Fall gekommen sind. Ein beeindruckendes Spektakel in welchem ein Stein nach dem anderen umgestoßen und mit einem Klirren am Boden aufschlug. Doch wie dieses Prinzip des Fallenden Dominosteins tatsächlich funktioniert, wissen nur die wenigsten. Mit genau diesem Grundsatz haben wir uns diese Woche beschäftigt und interessante Einsichten in dieses Thema erlangt. Die Formeln und Beschreibungen der Vorgänge wurden von uns selbst mit höchster Sorgfalt erarbeitet und deren Genauigkeit überprüft. Glücklicher Weise konnten wir die Woche gut nutzen und alle uns gestellten Problemstellungen lösen.

1.2. Problemstellung

Der Domino-Effekt ist eine Art Kettenreaktion, eine Abfolge von Ereignissen, bei der durch jedes einzelne Ereignis ein ähnliches weiteres Ereignis ausgelöst wird. Mit den Steinen des Domino Spiels kann man diesen Effekt am besten demonstrieren. Dabei werden die Steine hochkant so in einer Reihe aufgestellt, dass beim Umfallen eines Steines der nächste ebenfalls umgeworfen wird. Dies setzt sich fort, bis alle Steine umgefallen sind.

Offenbar ist dieser Vorgang faszinierend anzusehen. Alljährlich begeistert der Domino Day zahlreiche Fernsehzuschauer, wobei jedes Mal versucht wird, möglichst viele Steine effektiv für einen neuen Weltrekord zu arrangieren. So betrug 2009 die Anzahl der Steine bereits fast fünf Millionen.

Am 9.11.2009 wurde der 20. Jahrestag des Falls der Berliner Mauer gefeiert, indem eine eineinhalb Kilometer lange Reihe von ca. 1000 Riesendominosteinen umgeworfen wurde, die entlang des ehemaligen Mauerverlaufs aufgestellt waren.

Uns interessiert hier die Mathematik und Mechanik des Dominoeffekts. Wie schnell bewegt sich die "Umfallwelle" vorwärts? Kann man, mit einem kleinen Stein beginnend, am Schluss ein Hochhaus umwerfen? Wie steil kann eine Dominoreihe bergauf laufen? Fragen über Fragen, die auf eine genauere Analyse warten.

2.1. Geometrie

2.1.1. Parameter

Nach anfänglichem Brainstorming bezüglich der Problemstellung, haben wir uns überlegt welche, für uns relevanten Abmessungen vorhanden sind und sie in einer Skizze graphisch dargestellt.

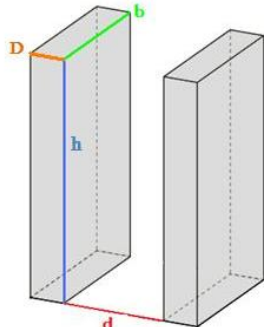
	D	Dicke	
	h	Höhe	$h > D$
	b	Breite	
	m	Masse	
	d	Distanz	$0 < d < h$

Abb. 1: Parameterskizze

2.1.2. Geometrische Betrachtung des Kippvorgangs eines Stein

Um den Kippvorgang näher erläutern zu können ist es nötig, ein Modell für die einzelnen Abschnitte des Kippens zu erstellen. Hierbei erwies sich die Vorstellung, dass eine Drehachse durch die untere rechte Kante verläuft und so den Stein fixiert, als äußerst zweckdienlich. Diese Annahme entspricht zumindest bei einem relativ rutschfesten Untergrund mit an Sicherheit grenzender Wahrscheinlichkeit der Realität.²

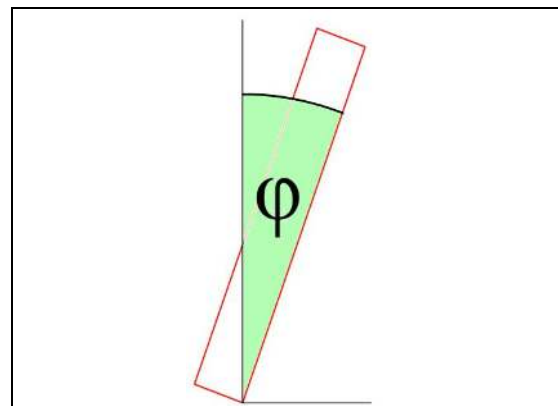


Abb.2: Neigungswinkel

Weiters werden wir im Folgenden den Kippvorgang über die einzelnen Kippwinkel veranschaulichen, was die Einführung einer neuen Winkelvariablen φ erfordert. Diese beschreibt den Winkel zwischen der Lotrechten und der Steinkante. Aus Gründen der besseren Anschaulichkeit haben wir uns dazu entschlossen, vorerst im zweidimensionalen Raum zu arbeiten.

² Als Hinweis darauf ist das Bildmaterial vom sogenannten „Domino Day“ zu betrachten. Dort ist es möglich, Bilder aus den umgefallenen Steinen zu erzeugen, was nur durch eine relativ geringe Abweichung vom ursprünglichen Standort erreicht werden kann.

Als besonderen Winkel wollen wir φ_k herausgreifen. Dieser beschreibt die Lage des Steins im kritischen Zustand, in dem schon die kleinste Einwirkung ein Fallen zur Folge hätte (= instabiles Gleichgewicht).

Dieser Winkel ist genau dann erreicht, wenn sich der Schwerpunkt des Steins exakt senkrecht über der Drehachse befindet, also der Stein gerade noch stehen bleibt.

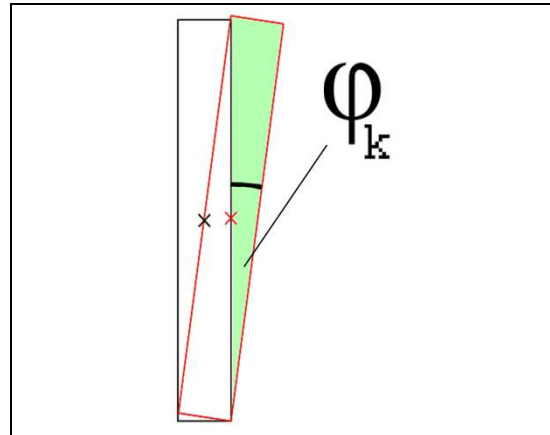


Abb.2: Kritischer Winkel

Für φ_k ergibt sich aus der Trigonometrie somit folgende Formel:

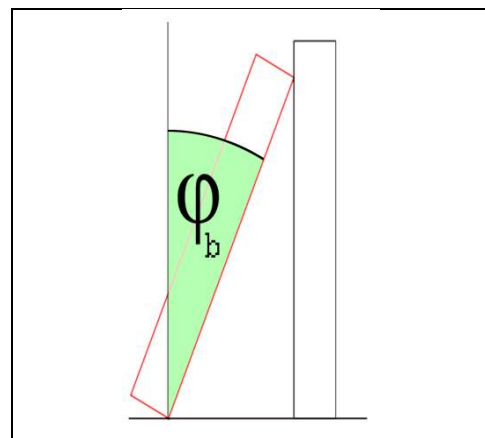
$$\varphi_k = \tan^{-1} \left(\frac{D}{h} \right)$$

2.1.3. Geometrische Betrachtung des Kippvorgangs mehrerer Steine

Sobald nun ein zweiter Stein oder gar eine ganze Reihe von Steinen in den Fallvorgang einbezogen wird, ergibt sich ein weiterer wichtiger Winkel: der Berührungswinkel φ_b .

Für diesen ergibt sich folgende Formel:

$$\varphi_b = \sin^{-1} \left(\frac{d}{h} \right)$$

Abb.3: Berührungswinkel φ_b

Mit dieser Formel der Winkelberechnung kann folgendes Modell einer fallenden Dominoreihe erstellt werden:

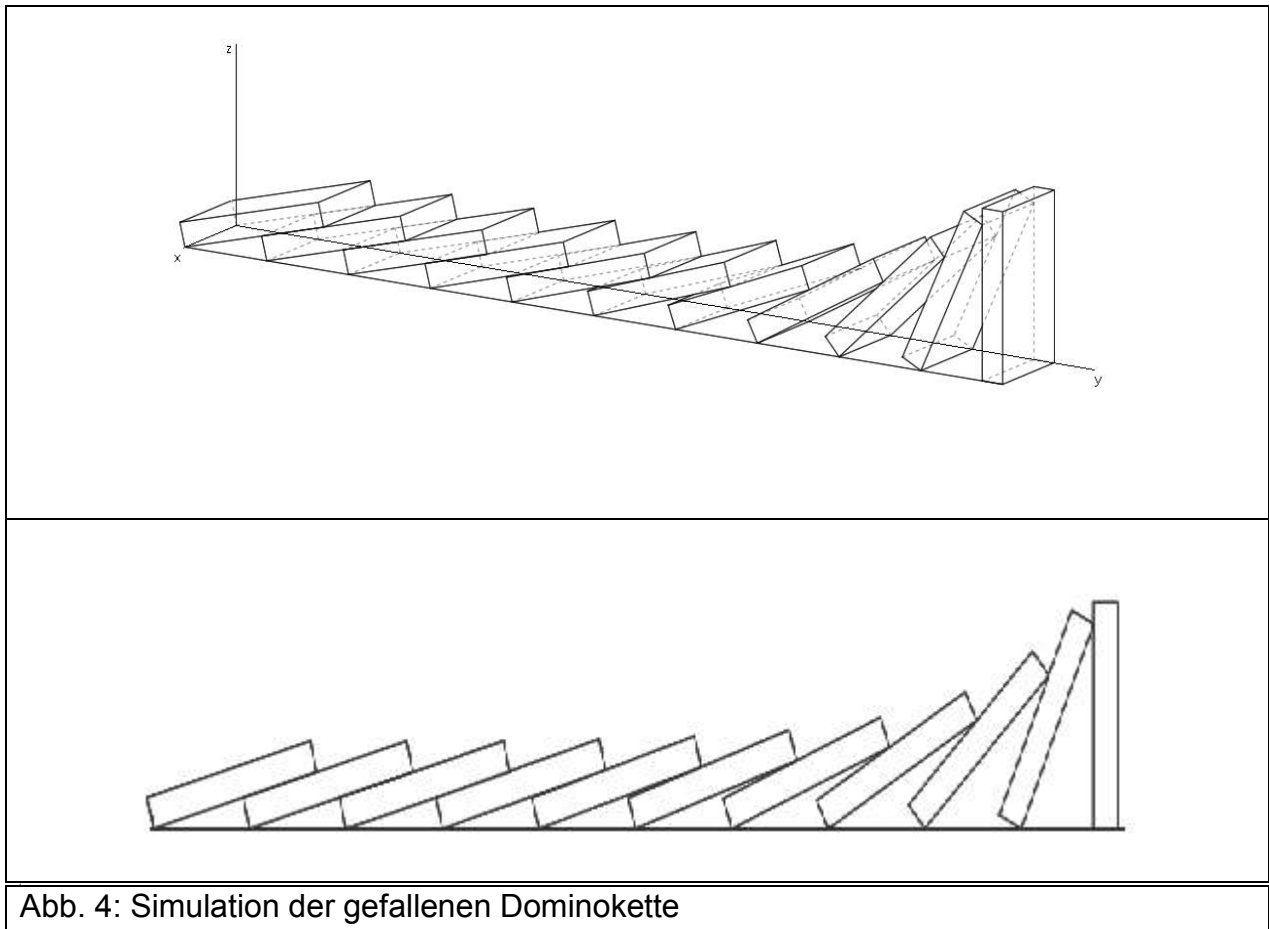


Abb. 4: Simulation der gefallenen Dominokette

In einer solchen Kette hängt φ_{n-1} jeweils vom Neigungswinkel des vorderen Steins φ_n ab:

$$\varphi_{n-1} = \varphi_n + \sin^{-1} \left(\frac{(D + d) * \cos(\varphi_n) - D}{h} \right)$$

2.2. Dynamische Größen

2.2.1 Drehmoment

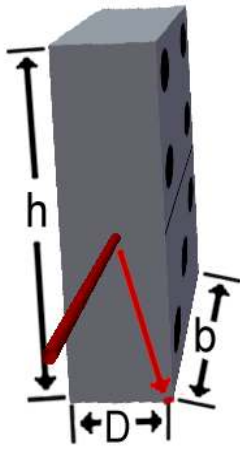
Nachdem wir uns näher mit dem Drehmoment beschäftigt hatten, unterschieden wir zwei Arten der unterschiedlichen Berechnung.

A) $M(\varphi(t)) = I \cdot \varphi''(t)$

Für das Berechnen der ersten Drehmomentformel benötigt man folgende Größen:

I	Trägheitsmoment
$\varphi''(t)$	2. Ableitung von φ (Winkelbeschleunigung) im Bezug auf die Zeit t

Um die folgenden Rechenschritte anschaulicher darzustellen zu können, haben wir das nachstehende Modell erstellt:

	$I_{\text{Körper}} = 1/12 \cdot m \cdot (h^2 + D^2)$
Abb.5: Ein Quader, mit verschobener Drehachse	

Wir verwendeten die oberhalb angeführte Formel zur Berechnung des Trägheitsmomentes, welche die Tatsache mit einbezog, dass die Drehachse durch den Schwerpunkt verläuft.

Um die Formel an unsere Ausgangssituation, in welcher die Drehachse eine andere Stellung einnimmt, anzupassen, verwendeten wir den Satz von Steiner.

Satz von Steiner: Dieser eignet sich dazu, die Trägheitsmomente, oder den Drehimpuls eines starren Körpers bezüglich einer Rotation um eine Drehachse zu berechnen, die nicht durch den Massenmittelpunkt des Körpers verläuft.

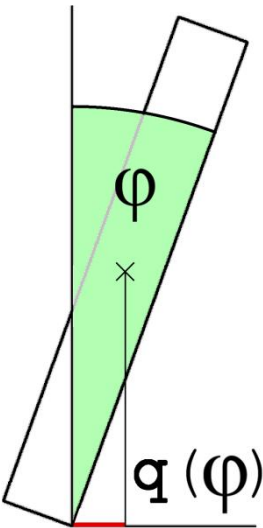
$$I = I_{\text{Körper}} + m \cdot l^2$$

l → halbe Diagonale $[\sqrt{(D^2+h^2)}/2]$	Die Strecke um die man die Drehachse verschieben will.
--	--

Daraus ergibt sich diese Formel: $I = 1/3 \cdot m \cdot (h^2 + D^2)$

B) $M(\varphi(t)) = m \cdot g \cdot q(\varphi(t))$

Für die Berechnung der zweiten Drehmomentformel benötigt man folgende Größen, die wir in einem entsprechenden Modell auch gleich bezeichnet haben:

g	Erdbeschleunigung
q(φ(t))	horizontaler Abstand von der Drehachse zur Senkrechten durch den Schwerpunkt im Bezug auf die Zeit
	
Abb.6: Dominostein in einem beliebigen Winkel φ(t)	

Um q(φ(t)) berechnen zu können, haben wir folgende Formel entwickelt:

$$q(\varphi(t)) = \frac{1}{2} \cdot \sqrt{D^2 + h^2} \cdot \sin(\varphi(t) - \varphi_k)$$

Anschließend an diese zwei Berechnungen für das Drehmoment haben wir diese beiden gleichgesetzt:

$$\frac{1}{3} m (D^2 + h^2) \varphi''(t) = \frac{m \cdot g}{2} \sqrt{D^2 + h^2} \sin(\varphi(t) - \varphi_k)$$

Wie wir festgestellt haben, ist die Masse kürzbar.

Daraus lässt sich schließen, dass bei Steinen mit der gleichen Masse, die Massen für das Drehmoment irrelevant sind. Eine Ausnahme davon bildet lediglich der Anfangsstein, für den die Masse aus Gründen **differenten Anfangsgeschwindigkeiten** sehr wohl den Ablauf der Kettenreaktion beeinflusst.

2.3 Energiebetrachtungen

2.3.1. Vorbemerkungen

Zu allererst bestimmten wir potentielle - und Rotationsenergie der Dominosteine.

E_{rot}	Rotationsenergie	$I \cdot \omega^2 / 2$
E_{pot}	Potentielle Energie	$m \cdot g \cdot h_{\text{SP}}$
ω	Winkelgeschwindigkeit	
h_{SP}	Höhe des Schwerpunktes	

Unsere nachfolgenden Berechnungen basieren auf dem Energieerhaltungssatz.

$$E_{\text{rot}} + E_{\text{pot}} = \text{konstant}$$

Durch Gleichsetzen der Formeln der beiden Energien und anschließendes Umformen erhielten wir eine Formel für $\omega(\varphi)$:

$$\omega(\varphi(t)) = \sqrt{\frac{h - \sqrt{h^2 + D^2} \cdot \cos(\varphi - \varphi_k) + I \cdot \omega(0)^2}{I}}$$

2.3.2. Fallzeit

Anhand der oben angeführten Funktion, lässt sich auf die Fallzeit für einen Dominostein schließen.

$$T(\varphi) = \int_0^{\varphi} \frac{d\varphi}{\omega(\varphi)}$$

Durch Einsetzen der Geschwindigkeitsfunktion in die Zeitfunktion, erhält man folgende Gleichung:

$$T(\varphi) = \int_0^{\varphi} \frac{d\varphi}{\sqrt{\frac{h - \sqrt{h^2 + D^2} \cdot \cos(\varphi - \varphi_k) + I \cdot \omega(0)}{I}}}$$

An sich brauchbar, jedoch für weiterführende Berechnungen zu komplex, erscheint es wenig zweckdienlich, weitere Dominosteine in diese Funktion mit einzubeziehen. Bereits die Miteinbeziehung eines einzigen weiteren Steines, ließe die Formel wie folgt aussehen:

$$T(\varphi) = \int_0^{\varphi} \frac{m \cdot (h^2 + D^2)}{6 \cdot E_{kin(anfang)} + 6 \cdot mg \left(\frac{h}{2} - \sqrt{h^2 + D^2} \cdot \cos(\varphi - \varphi_k) \right) + 6 \cdot \int_0^{\varphi} \left[\frac{\sqrt{h^2 + D^2} \cdot \sin(\varphi_n + \sin^{-1} \left(\frac{(D+d) \cdot \cos(\varphi_n) - D}{h} \right) - \varphi_k) \cdot mg \cdot \cos \left(\varphi_n + \sin^{-1} \left(\frac{(D+d) \cdot \cos(\varphi_n) - D}{h} \right) \right)}{2 \cdot \cos(\varphi_n + \sin^{-1} \left(\frac{(D+d) \cdot \cos(\varphi_n) - D}{h} \right) - \varphi_n)} \right] d\varphi} d\varphi$$

Jedoch war es uns möglich aufgrund eines Experiments eine Aussage über die Fallgeschwindigkeit zu machen.

→ Je kleiner der Abstand zwischen den Steinen ist, desto schneller ist ihre Fallbewegung.

2.3.3. Zusammenhang der Winkelgeschwindigkeiten

ω_1	Winkelgeschwindigkeit des stoßenden Steines bei der gemeinsamen Bewegung
ω_2	Winkelgeschwindigkeit des gestoßenen Steines bei der gemeinsamen Bewegung

Die Winkelgeschwindigkeit eines Steines ω_1 wird immer durch die Winkelgeschwindigkeit des davorliegenden Steines ω_2 bestimmt.

Dabei erhält man ω_1 indem die Funktion des Winkels φ nach φ abgeleitet wird und mit der Winkelgeschwindigkeit ω_2 multipliziert wird.

$$\omega_1(t) = \left. \frac{df(\varphi)}{d\varphi} \right|_{\varphi=\varphi_2(t)} \cdot \omega_2(t)$$

ω_2 ergibt sich durch Ableiten des Winkels φ_2 (Winkel des gestoßenen Steines) nach der Zeit t .

$$\omega_2(t) = \frac{d\varphi_2(t)}{dt}$$

2.3.4. Impulserhaltung beim Stoß

ω_1	Endwinkelgeschwindigkeit des 1. Steines bei der Berührung (φ_b)
ω_2	Anfangswinkelgeschwindigkeit der gemeinsamen Bewegung

Durch die Betrachtung des Impulserhaltungssatzes und die Annahme, dass die Horizontalkomponente des Impulses der Schwerpunktbewegung erhalten bleiben soll, ergibt sich folgende Formel für die Anfangswinkelgeschwindigkeit der gemeinsamen Bewegung der beiden Steine:

$$\omega_2 = \frac{\omega_1 \cdot \cos(\varphi_b - \varphi_k)}{\cos(\varphi_k) + \left. \frac{df(\varphi)}{d\varphi} \right|_{\varphi=0} \cdot \cos(\varphi_b - \varphi_k)}$$

2.4 Steigung

Ein weiterer Aspekt unserer Problemstellung bezieht sich auf die Steigung des Untergrunds. Da es uns aufgrund von Unklarheiten in Bezug auf die Kraftübertragung von einem Stein auf den nächsten nicht möglich ist, ein korrektes mathematisches Modell zu erstellen, beschränken wir uns auf theoretische Annäherungen an das Problem.

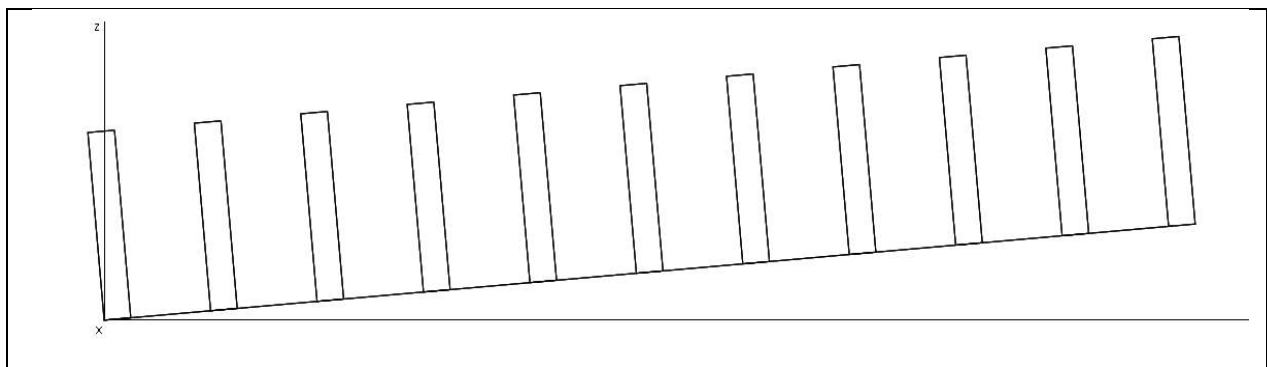
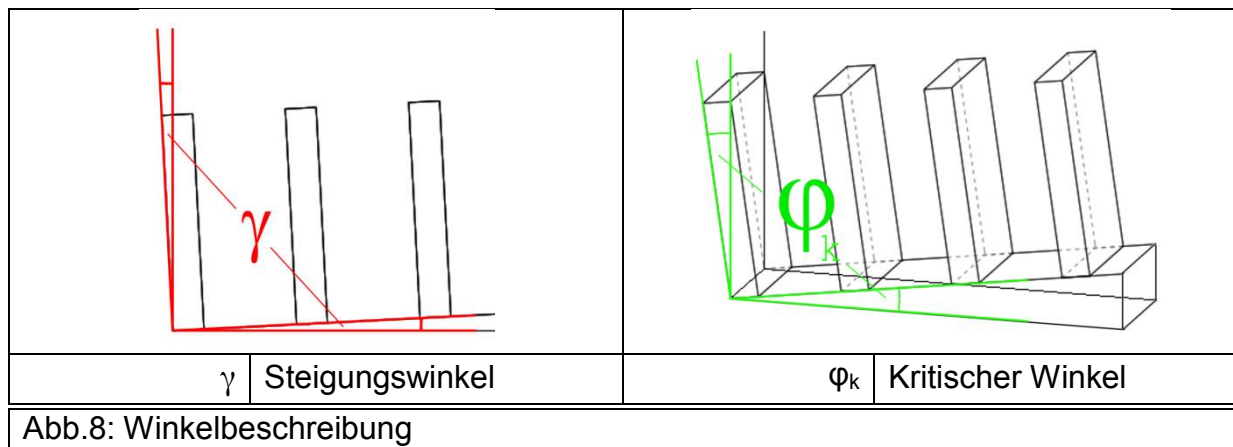


Abb.7: Dominoreihe mit ansteigendem Untergrund

2.4.1. Maximale Steigung

Um das Aufstellen der Steine überhaupt erst zu ermöglichen, darf der Steigungswinkel γ den vorher beschriebenen Winkel ϕ_k nicht überschreiten. $\rightarrow \gamma_{\max}$

E_{nec}	Benötigte Energie
E_{w}	Weitergegebene Energie

Eine Voraussetzung für das Fallen ist, dass E_{w} größer als E_{nec} ist. $\rightarrow E_{\text{w}} > E_{\text{nec}}$

Damit die Energie eines Steins ausreicht um seinen Nachfolger umzukippen muss E_{nec} groß genug sein, um die beiden Winkel γ , ϕ_k zu überwinden.

Die Formel für die Energie die aufgebracht werden muss lautet wie folgt:

$$E_{\text{nec}} = \left[\sqrt{h^2 + D^2} - \sqrt{h^2 + D^2} \cdot \cos(\gamma + \phi_k) \right] \cdot mg$$

2.6. Exponentielles Wachstum

Nach der Beschreibung der grundlegenden dynamischen Eigenschaften eines sowie mehrerer Dominosteine mit gleicher Größe und Gewicht, wollen wir uns nun der mathematischen Darstellung einiger kurioser Gedankenexperimente zu diesem Thema widmen.

Ein naheliegendes Beispiel dafür ist wohl die Auswirkung der Veränderungen von Größe und Masse der Dominosteine.

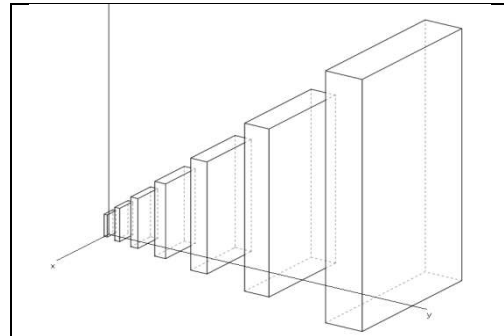


Abb.9:exponentielles Wachstum

Ein gewöhnlicher Dominostein der Höhe 1 vermag einen weiteren der Höhe 1,5 zu Fall zu bringen. Führen wir das Experiment fort, so ergibt sich für die Größe des Steins ein exponentielles Wachstum in folgender Art:

Dominostein:	Höhe:
1. Stein	4,9cm
2. Stein	7,4cm
3. Stein	11,0cm
4. Stein	16,5cm

Daraus lässt sich eine Formel³ für die Größe bzw. Höhe des n-ten Steins erstellen:

$$h_n = h_1 \cdot 1,5^{n-1}$$

Diese Gleichung ist auch auf Breite, Dicke und Distanz anwendbar.

Im Gegensatz dazu wächst die Masse von einem Stein zum nächsten um den Faktor $1,5^3$ bzw. um den Faktor 3,375.

Um die Auswirkungen dieser Erkenntnisse zu verdeutlichen, haben wir es an einem praktischen Beispiel veranschaulicht.

³ Online im Internet: URL:

http://books.google.at/books?id=iN0HQ9E_YT4C&pg=PA34&lpg=PA34&dq=dominosteine+fallgeschwindigkeit&source=bl&ots=EoID0UYoV1&sig=pGLJhF8GMgw4eH7mvKu-IKPtHrY&hl=de&ei=IC51S63AHKf-sQa15oyXCg&sa=X&oi=book_result&ct=result&resnum=3&ved=0CBAQ6AEwAg#v=onepage&q=&f=false

Stand von 12.02.2010

Hierfür haben wir uns einen Stein in der Höhe des höchsten Gebäude der Welt, dem Burj Khalifa in Dubai vorgestellt, denn wir umzuwerfen gedenken.

Unser Ziel ist es herauszufinden, wie viele Steine benötigt werden um diesen Stein zu Fall zu bringen.

h_n	Höhe des Burj Khalifa	828 m
h_1	Höhe des Anfangssteines	0,049 m

$$828 = 0,049 \cdot 1,5^{n-1}$$

$$\log_{1,5} \frac{828}{0,049} = n - 1$$

$$n = 25$$

→ Bereits der 25ste Stein hat die Höhe des Burj Khalifa. Es werden daher 24 Steine benötigt diesen umzuwerfen.



Abb.10: Burj Khalifa in Dubai im Vergleich mit dem Eiffelturm

Wie bereits erwähnt, beträgt der Wachstumsfaktor von einem Stein zum nächsten nicht 1,5 sondern $1,5^3$.

Daher lautet die Formel für die Berechnung der Masse wie folgt:

$$m_n = m_1 \cdot 1,5^{3(n-1)}$$

Auf unser Beispiel bezogen würde dies so aussehen:

m_1	Masse des ersten Steines	
m_n	Masse des letzten Steines	?

$$m_{25} = m_1 \cdot 1,5^{3(25-1)}$$

$$m_{25} = 53\,000\,000 \text{ t}$$

→ Dies entspricht ungefähr dem Gewicht von 8,2 Millionen Elefanten



Prof. Georg Propst

Nachhaltige Forstwirtschaft

Ökologie

Sara Basara, Thessa Hinteregger,
Stefan Müller, Milena Posch,
Benedikt Schönhuber, Philipp Wakonigg

Problemstellung:

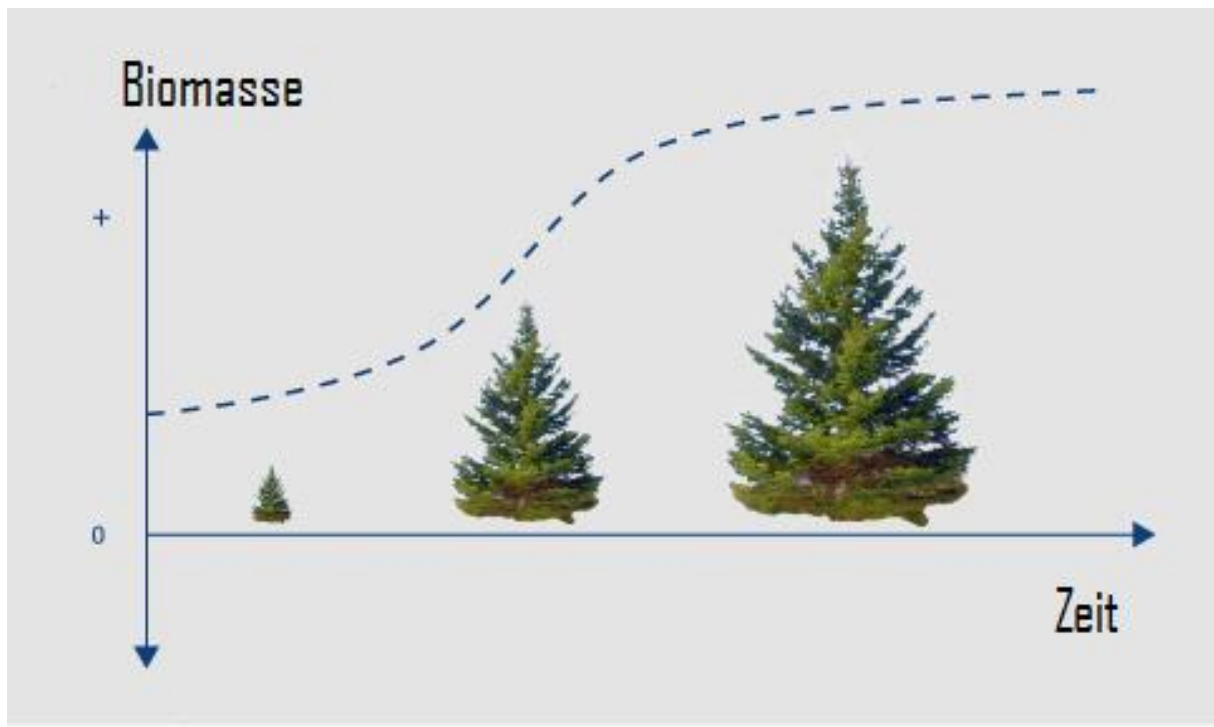
Der Begriff Nachhaltigkeit wurde erstmals in der Forstwirtschaft verwendet und bezeichnet eine Strategie der Entnahme/Nichtentnahme und Aufforstung, die den Bestand nicht gefährdet, sondern stabil lässt oder vermehrt. Diese Strategien sollen unter Umständen auch aktuelle Entwicklungen berücksichtigen, wie zum Beispiel Sturmschäden und Borkenkäferbefall.

In diesem Projekt hatten wir mathematische Modelle entworfen, die die zeitliche Entwicklung eines bewirtschafteten Waldbestandes beschreiben und simulieren. Es war eine breite Palette von Modellen denkbar, etwa räumlich homogen oder strukturiert, mit gleichbleibenden oder altersabhängigen Wachstumsraten, mit Zerfallselementen, saisonale Effekten, Ausbreitung von Schädlingen, etc.

Anhand der Modelle konnten wir mögliche ökologische und ökonomische Auswirkungen verschiedene Entnahme- und Aufforstungsstrategien untersuchen. Die dabei auftretenden Fragestellungen und Methoden wurden von uns selbst gewählt.

Das Ziel unseres Projektes war daher das Modellieren möglicher ökologischer und ökonomischer Auswirkungen verschiedener Forstwirtschaftsstrategien zu untersuchen und darzustellen. Das von uns verwendete Programm für diese Modellierung war MATLAB.

MODELLIERUNG:



Quelle: www.firstforest.de/Wertentwicklung.jpg

Zu Beginn unserer Überlegungen legten wir fest, dass uns eine nicht bewirtschaftete Fläche zur Verfügung steht. Auf dieser Fläche A gibt es Platz für x Bäume ($x = A/4r^2$) mit dem Radius r , der mit der Zeit t wächst. Da in der Natur Wachstumsprozesse meistens zuerst exponentiell verlaufen und anschließend durch eine Kapazitätsgrenze C , die das Erliegen des Wachstums darstellt, beschränkt werden, nützten wir diese Idee und entwickelten folgende Formel, die durch die obere Grafik bestätigt wurde:

$$G(t) = 2C \cdot \frac{e^{z(t-t_0-\Delta t)}}{e^{z(t-t_0-\Delta t)} + 1}$$

$G(t)$... Biomasse in Tonnen zum Zeitpunkt t (in Jahren)

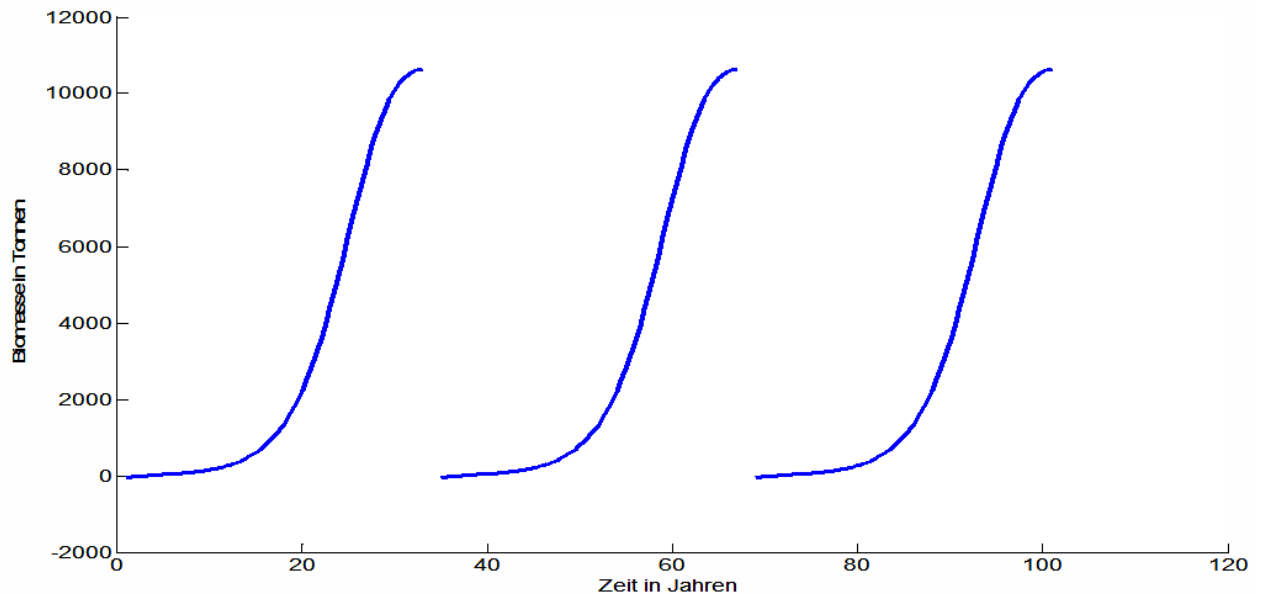
$2C$... Maximale Biomasse des Baumes in Tonnen

e ... Euler'sche Zahl

z ... Konstante der Wachstumsgeschwindigkeit der jeweiligen Baumart

t_0 ... Zeitpunkt, zu dem der Baum gepflanzt wird

Δt ... Zeitraum von t_0 bis zum stärksten Wachstum des Baumes



Doch vor dieser endgültigen „Baumformel“ hatten wir noch einige andere Ansätze, wie zum Beispiel:

1)

$$0 < C \frac{e^{at} - 1}{e^{at} + 1} = \frac{1 - e^{-at}}{1 + e^{-at}} \xrightarrow{t \rightarrow \infty} 1$$

Durch diese Formel wollten wir verhindern, dass das Wachstum gegen 0 geht.

2)

$$C(1 - e^{-t})$$

Aufgrund eines Tests, den wir während der Modellierungswoche durch zu führen hatten, kamen wir auf diese Formel. Doch wir haben sie wieder verworfen, weil die Steigung zu Beginn zu steil und linear verlief.

Im Zuge unserer Überlegungen kamen wir zu dem Schluss, dass eventuell eine „Heaviside-Funktion“ uns weiterhelfen könnte.

Erklärung:

Die „Heaviside-Funktion“ nimmt für jede beliebige negative Zahl den Wert 0 an, für positive den Wert 1. Sie ist also eine Funktion für positive reelle Zahlen.

Jedoch haben wir diese nicht weiter verwendet, da wir unsere Funktionen so umgeformt haben, dass sie niemals negativ sein können.

Zusätzlich errechneten wir durch den „Spezialfall“ $G_{1(0)} \Delta t$ da C , z und t_1 gegeben waren und Δt in unserem Modell eine gleichbleibende wichtige Konstante darstellt:

$$t_0 = t_1 - \Delta t$$

$$G_{1(0)} = C \frac{2e^{z(-\Delta t)}}{e^{z(-\Delta t)} + 1}$$

$$\Rightarrow \Delta t = \frac{1}{z} [\ln(2C - G_{1(0)}) - \ln G_{1(0)}]$$

Hierbei bedenken wir auch die Richtigkeit unserer einzelnen Variablen und endeten schließlich damit, den Wendepunkt in t_1 zu beweisen:

$$G'_{1(t)} = 2Cz \cdot \frac{e^{(z(t-t_1)+1)}}{(e^{(z(t-t_1)+1)})^2} = 2Cz^2 \cdot \frac{(e^{(z(t-t_1)} + 1 - 2 \cdot e^{z(t-t_1)})}{e^{(z(t-t_1)+1)^3}} \cdot e^{z(t-t_1)}$$

$$G'_{1(t)} = \frac{2Cz^2 e^{z(t-t_1)}}{(\dots)^3} \cdot (1 - e^{z(t-t_1)})$$

$$\Rightarrow G'_{1(t)} \gtrless 0 \Leftrightarrow \begin{matrix} t < t_1 \\ t = t_1 \\ t > t_1 \end{matrix}$$

Daraus können wir folgende Schlussfolgerungen ziehen:

- t_1 ist tatsächlich ein Wendepunkt
- Vor t_1 ist die Funktion konkav
- Nach t_1 ist die Funktion konvex

Nun wollten wir aber auch das potentielle Einkommen eines Försters berechnen. Zu diesem Zweck haben wir uns über Einkaufs- und Verkaufspreise Gedanken gemacht und überlegt, wie man diese bestmöglich mit unserer bestehenden Funktion vereinbaren könnte:

$$EJ(t) = \frac{(pv \cdot G(t) - G_0 \cdot pe)}{t - t_0 + pause}$$

EJ... durchschnittliches Einkommen pro Jahr

t... Zeitpunkt(in Jahren)

EJ(t)...das Einkommen bis zum Jahr t durch die Anzahl der vergangenen Jahre

pv... Verkaufspreis pro Tonne Biomasse

G(t)... Biomasse in einem bestimmten Jahr (siehe oben)

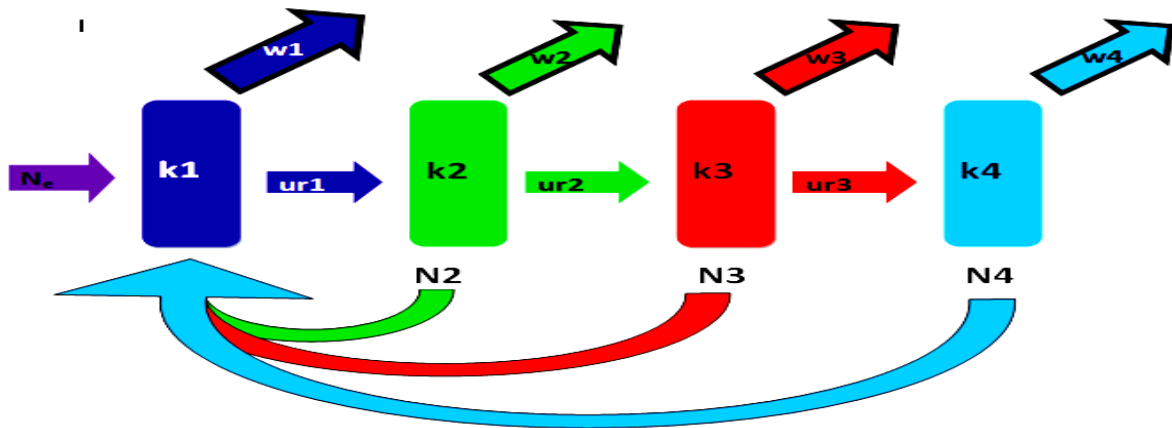
G_0 ... Biomasse in Tonnen, mit der der Baum angepflanzt wird

pe... Einkaufspreis pro Tonne Biomasse

t_0 ... Zeitpunkt, zu dem der Baum gepflanzt wird

pause...Anzahl an Jahren, die nach dem Fällen des Waldes vergehen, bevor neue Bäume gesetzt werden.

Um eine Struktur in unseren bewirtschafteten Wald zu bringen führten wir die Idee einer Gruppeneinteilung ein, die zuerst in Höhe und Alter unterteilt war. Jedoch stellten wir nach einiger Zeit fest, dass die Biomasse eine wichtigere Rolle für die Forstwirtschaft spielt. Daher entwickelten wir ein neues System:



Unsere Überlegung zu diesem Modell lautet wie folgt:

Es existieren 4 Klassen $k_{1(t)}, k_{2(t)}, k_{3(t)}, k_{4(t)}$, die jeweils nach ihrer durchschnittlichen Biomasse m_1, m_2, m_3, m_4 eingeteilt und identifiziert werden können. In der ersten Klasse $k_{1(t)}$ befinden sich Bäume der geringsten Masse m_1 , usw., bis hin zur letzten mit Bäumen der höchsten Masse m_4 . Bäume der ersten Klasse können sich noch nicht fortpflanzen ($N_1 = 0$), ab der zweiten Klasse ist dies für jene möglich und so bringen sie Nachwuchs N_2, N_3, N_4 von ihrer zur ersten Klasse. Beachtet werden muss bei diesem Modell, dass die Übergangsrate ur von der jeweiligen Klasse abhängt und aus der Funktion G_1 ablesbar ist. Das bedeutet, dass die Übergangsraten in den mittleren Klassen, die t_1 am nächsten sind, am höchsten sind. Zusätzlich besteht die Möglichkeit, durch sog. externen Nachwuchs N_e (ansetzen) die Stabilität der Forstwirtschaft zu gewähren. Der Förster entnimmt je nach Klasse einen gewissen Anteil an Bäumen w_1, w_2, w_3, w_4 , wobei berücksichtigt werden muss, dass der letzten Klasse am meisten Bäume entnommen und anschließend verkauft werden.

Das dazugehörige Gleichungssystem kann man sich wie folgt vorstellen:

$$k_1(t+1) = k_1(t) - ur_1 k_1$$

$$k_2(t+1) = k_2(t) - ur_2 k_{1t} + ur_1 k_{1t}$$

$$k_3(t+1) = k_3(t) - ur_3 k_{3t} + ur_2 k_{2t}$$

$$k_4(t+1) = k_4(t) - ur_4 k_{4t} + ur_3 k_{3t}$$

Im Programm MATLAB wird allerdings mit Matrizen gearbeitet; deshalb mussten wir dieses Gleichungssystem als Matrix angeben zu der wir dann auch den Nachwuchs hinzugefügt haben.

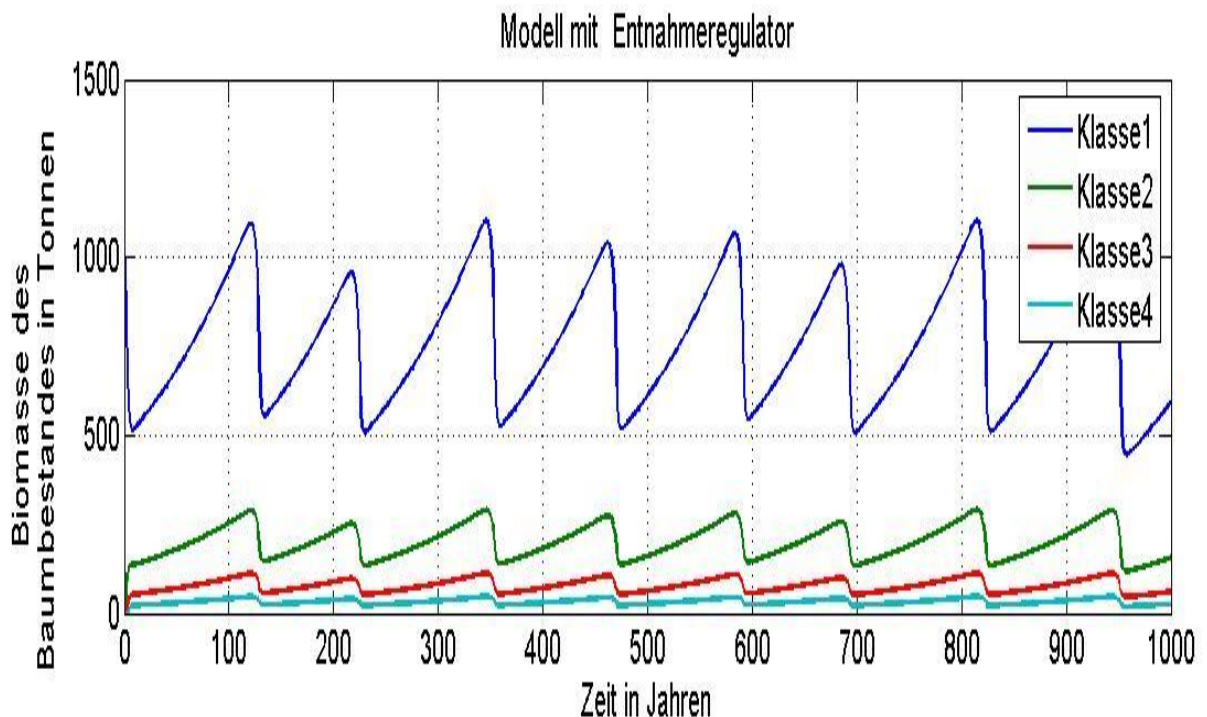
$$\begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{pmatrix}_{t+1} = \begin{pmatrix} 1 - ur_1 - w_1 + N_1 & N_2 & N_3 & N_4 \\ ur_1 & 1 - ur_2 - w_2 & 0 & 0 \\ 0 & ur_2 & 1 - ur_3 - w_3 & 0 \\ 0 & 0 & ur_3 & 1 - ur_4 - w_4 \end{pmatrix} \begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{pmatrix}_t + \begin{pmatrix} Ne \\ 0 \\ 0 \\ 0 \end{pmatrix}_t$$

- N...Nachwuchs
- ur...Übergangsrate
- w...Entnahme
- k...Anzahl der Bäume
- Ne... externer Nachwuchs

Um mit Hilfe der Matrix das Gleichungssystem nachvollziehen zu können multipliziert man die erste Zeile der Matrix mit dem Spaltvektor k, dann die zweite Zeile etc.

Bei Simulation mit dieser Matrix und Ne=0 ist nie ein anderer Fall aufgetreten als exponentielles unbegrenztes Wachstum oder Abfall gegen 0. Um auch nicht triviale Gleichgewichte zu erzeugen wählten wir die Parameter so, dass mit Ne=0 der Wald ausstirbt (gegen 0 geht). Mit Ne> 0 kann man dann erreichen, dass die Klassenanzahlen gegen konstante positive Werte streben.

Durch unterschiedliche Steuerung der Entnahme durch Beobachtung des Waldbestands versuchten wir, einen Ausgleich zu erzielen- was zu Schwankungen um einen stabilen Mittelwert führte.



In MATLAB haben wir das mit Hilfe folgenden if- Befehls ausgeführt:

```
if k(1) > 1050;  
    w(w~=0)= w(w~=0)+ deltax  
elseif k(1)< 950  
    w(w~=0)= w(w~=0)- deltax  
end
```

Um ein möglichst realistisches Modell zu entwickeln berücksichtigten wir auch Einflüsse durch die Umwelt und vor allem die Auswirkungen durch Licht. Da der Wald mit zu vielen Bäumen nicht mehr die Chance hat, sich auf bestem Wege zu entwickeln und die Bäume erster Klasse zu wenig Licht bekommen muss zunehmend mehr entnommen werden um ein Gleichgewicht herzustellen.

Ein weiterer Punkt, der von uns in diesem Modell berücksichtigt wurde, sind Schädlings- und Unwetterschäden. Ein Problem, das sich bei der Berücksichtigung von ebensolchen Schäden ergibt, ist, dass Schädlinge nicht nach einer besonderen Regel den Waldbestand zerstören, ebenso wenig gibt es ist es möglich mit Sicherheit einen Sturm schon Jahre im Voraus vorherzusehen.

In diesem Modell verwenden wir also für Schädlingsbefall und Sturmschäden eine Zufallsvariable , die, wenn sie in ein bestimmtes Intervall von Werten fällt, die Schädlinge und Sturmschäden „aktiviert“, sollte die Zufallsvariable nicht in diesem Intervall sein, passiert nichts, der Wald wächst nach den übrigen Regeln. Rechnerisch haben wir diese Idee verwirklicht, indem man die Matrix für das Klassensystem entsprechend, unter Verwendung der Zufallsvariablen erweitert.

Sturmschäden und Bäume die von Schädlingen befallen sind, werden noch im selben Jahr entfernt und um einen Bruchteil des übrigen Verkaufspreises verkauft.

Unser Endergebnis schaut in unserem M-File in MATLAB wie folgt aus:

```
clear all
close all

% Baummodell- Mittwoch
C=2;
M=C*[0.25 0.75 1.25 1.75];
pv=100;
pe=200;
pvb=30;
pvs=20;
% Nachwuchs
N= [0 0 2 3];
% Nachwuchs- extern
Ne=[20 0 0 0 ]';

tend= 100;
%Alokieren
K=zeros(4,tend);
EJ=zeros(1,tend);
% Übergang
ur= [ 1/13 10/35 1/3];
% Anteil an entnommmenen Bäumen
w= [0.1 0.1 0.5 0.9];
% Anfangszustand
kO= [1000 0 0 0]';
%Ausgaben
E=(-1)*(M*kO*pe);size(E(1));
EJ(1)=E;
clear K
% DIE MATRIX
Y=[1-ur(1)-w(1)+N(1) N(2) N(3) N(4); ur(1) 1-ur(2)-w(2) 0 0
    0 ur(2) 1-ur(3)-w(3) 0 ; 0 0 ur(3) 1-w(4)];
% Klassen(Matrix)
k= kO;
K(:,1)= k;
sumk= sum(k);

hold on

for t=2:tend;

% Borkenkäfer und andere Schönhuber'schen Schädlinge
X=rand(1);
Z=0.985;
if X>Z;
    B=X*0.75;
else
    B=1;
end
```

```

S=rand(1);
T=0.985;
if S>T;
    S=S;
    U=S*0.75;
else
    U=1;
end

% DIE MATRIX
Y=[B-(1-U)-ur(1)-w(1)+N(1) N(2) N(3) N(4); ur(1) B-(1-U)-ur(2)-w(2) 0 0
    0 ur(2) B-(1-U)-ur(3)-w(3) 0 ; 0 0 ur(3) B-(1-U)-w(4)];
%Einnahmen/Ausgaben
E=E+(w*(k.*M')*pv-sum(Ne)*pe)-(1-B)*(M*k)*pvp-(1-U)*(M*k)*pvs;
EJ(t)=E/t;

K(:,t)= k;
deltaw= min(w(w~=0))*1/20;
if sum(k)> 1050;
    w(w~=0)= w(w~=0)+ deltax
elseif sum(k)< 950
    w(w~=0)= w(w~=0)- deltax
end
sumk= sum(k)
% Licht= Freiraum/ Freiraum+ Nachwuchs * die Anzahl zum Zeitpunkt t
FR= 9000;
c= FR/(FR+N*k);
k= Y*k.*c +Ne;

if B<1;

figure(1)
plot(t,K(:,t),'o')

elseif U<1;

figure(1)
plot(t,K(:,t),'+')

end

end

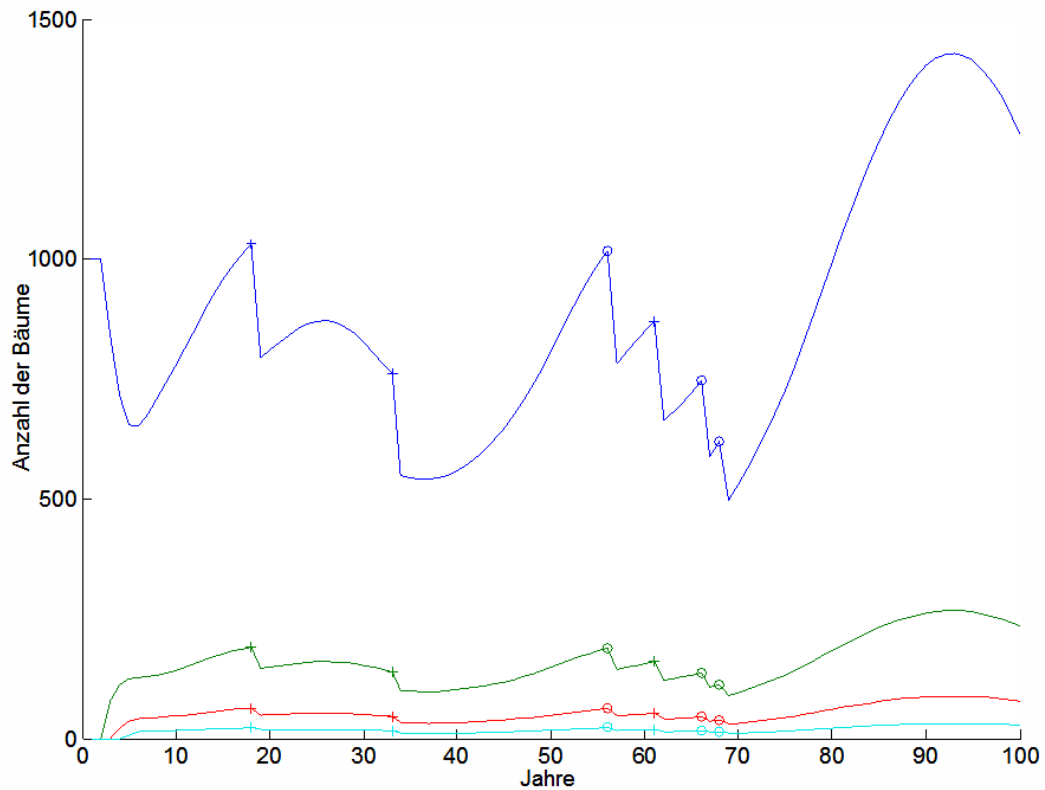
figure(1)
plot(1:tend,K)
title('Klasse1 = blau; Klasse2= grün; Klasse3= rot; Klasse4= hellblau ')
xlabel('Jahre')
ylabel('Anzahl an Bäumen in den Klassen')

figure(2)
plot(1:tend,EJ)

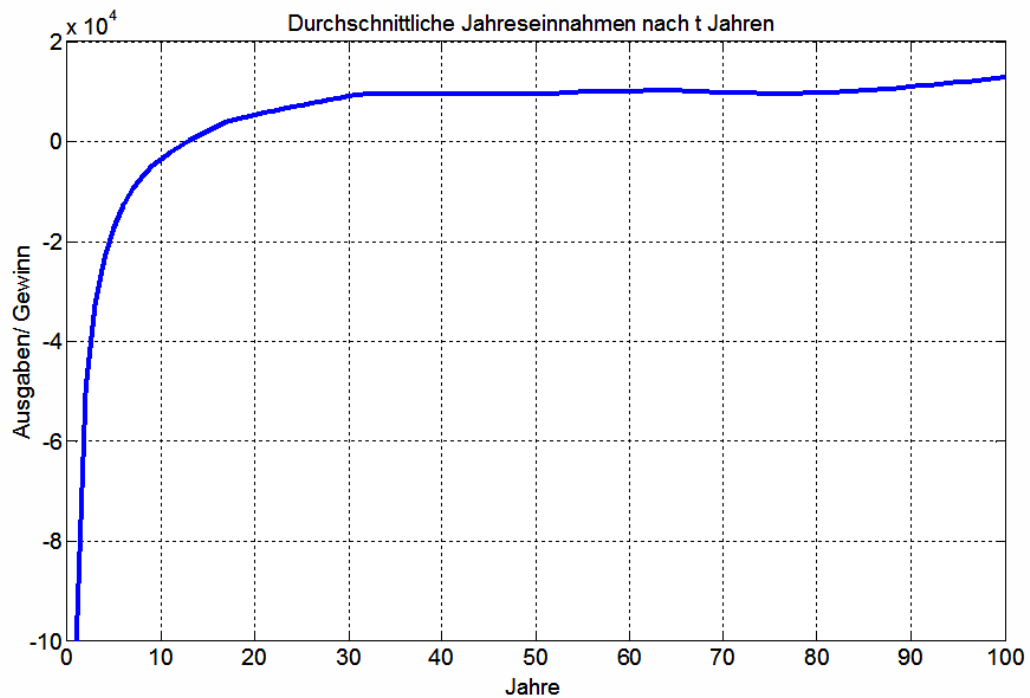
```

```
title('Durchschnittliche Jahreseinnahmen nach t Jahren')
grid on
```

Wird dieser File „geplottet“, erhalten wir diese Grafiken:



Hier ist unsere endgültige Strategie für eine nachhaltige Forstwirtschaft zu sehen, wobei die einzelnen Gruppen in verschiedene Farben (1. Klasse = blau, 2. Klasse = rot, 3. Klasse = grün, 4. Klasse = türkis) unterteilt sind. Dabei erkennt man, dass die 1. Klasse am zahlreichsten vorhanden ist und, dass sich der Wald selbst nach Sturmschäden (+) und Borkenkäferbefall (o) wieder erholt.



In dieser Grafik wird das durchschnittliche Jahreseinkommen eines Waldbesitzers veranschaulicht, wobei besonders die ersten 10 Jahre zu beachten sind. Das hohe Minus zu Beginn ist aufgrund des erforderlichen Einkaufs der Bäume zu erklären, die eingesetzt werden müssen; doch nach einiger Zeit pendelt sich das Ganze bei einem Betrag von 1000 €/Jahr ein.

Zusammenfassung:

Forstwirtschaft und Mathematik sind zwei Begriffe die selten zusammen verwendet werden.

Üblicherweise denkt niemand an Mathematik wenn von Strategien für nachhaltige Forstwirtschaft die Rede ist. Doch in dieser Woche lernten wir, dass auch diese Begriffe sehr wohl etwas miteinander zu tun haben können.

Als wir unsere Problemstellung zum ersten Mal hörten, hatten wir noch keine konkrete Vorstellung, wie wir diese Strategien modellieren könnten. Deshalb bekamen wir eine kurze Einführung von Prof. Georg Propst zu diesem Thema. Zu Beginn entwickelten wir ein Strategie die wir weiter auszubauen und zu bearbeiten versuchten.

Weiteres lasen wir Artikel zu diesem Thema, lernten mit MATLAB umzugehen, um unsere Aufforstungsstrategien modellieren zu können und hatten dabei mit der ein oder anderen Schwierigkeit zu kämpfen, jedoch wurden diese alle zeitgerecht gelöst. Trotz der nicht allzu leichten Problemstellung hatten wir beim Modellieren eine große Freude.

Schließlich und endlich möchten wir uns bei unserem Betreuer, Prof. Georg Propst für seine großartige Unterstützung bedanken, ohne die wir wohl kaum im Stande gewesen wären diese Simulation zu modellieren.

Vielen Dank.

Segmentierung von Bildern

Projekt: Bildverarbeitung

Betreuer: Prof. Stephen Keeling

ModelliererInnen: Bettina Wretschko, Elisabeth Hengge, Lukas Grabenwarter, Michael Reichelt, Martin Unger, Simeon Kanya

Problemstellung:

Das menschliche visuelle System erkennt Objekte im Sehfeld trotz verrauschter Interferenz und nicht gleichmäßiger Beleuchtung erstaunlicherweise gut. Wenn aber viele Objekte in einem Bild oder in mehreren Bildern im Detail identifiziert werden müssen, ist es nicht mehr praktisch, dass diese manuell bearbeitet werden. Zum Beispiel, wie könnte man die Organe in einer großen Zeitserie von Magnetresonanzbildern trennen und individuell untersuchen? Wenn der Kontrast oder die magnetische Erregung nicht gleichmäßig sind, ist die Intensitätsverteilung entsprechend irregulär. Wenn der Kontrast oder die magnetische Erregung nicht regelmäßig sind, ist die Intensitätsverteilung entsprechend niedrig. Um die interessierenden Objekte automatisch zu identifizieren, muss man dem Computer ein Rezept geben, wobei eine gegebene Menge von Intensitäten in Teilmengen oder Segmente geteilt wird.

Ziel:

Das Ziel des Projektes ist, ein Verfahren zur Segmentierung von realistischen Bildern zu entwickeln und dieses Verfahren am Computer zu implementieren.

Einleitung

Schon zu Beginn bekamen wir zwei Bilder, die wir im Laufe der Woche segmentieren sollten. Davon war eines ein computergeneriertes Musterbeispiel, das „Phantom“. Dessen Segmente hatten größtenteils verschiedene Intensitäten, wodurch sich die Segmentierung ungleich leichter gestalten sollte als bei dem zweiten Bild. Dieses ist ein realistisches MR-Bild, bei dem die Auflösung herabgesetzt wurde. Das primäre Ziel war es, die Leber durch Segmentierung herauszuheben, die Teilgruppen haben sich aber auch andere Ziele gesetzt und die meisten davon erreicht. Neben der Hauptaufgabe haben wir uns mit verschiedensten Problemen beschäftigt, wie dem Herausfiltern eines Rauschens oder dem Ausgleichen eines Farbverlaufs, aber auch mit der korrekten Segmentierung des Phantoms. Die Teilgruppen



Abb. 1 – „Phantom“

haben viele Ziele erreicht, doch nicht alle Lösungen waren optimal.

Einige Problemstellungen und Lösungsansätze:

- Segmentierung des Phantoms durch Konturenfindung und Vektoren (Lukas)
- Allgemeine Segmentierung (Martin & Simeon)
- Kontrastieren und Glätten (Michael)
- Interaktive Segmentierung (Elisabeth & Bettina)

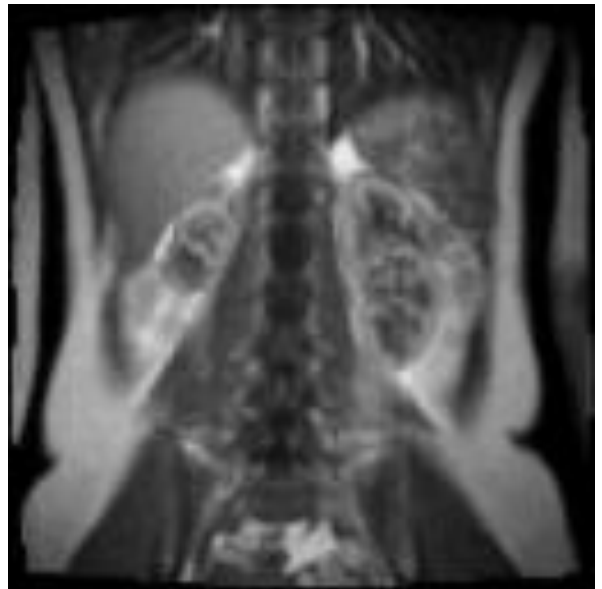
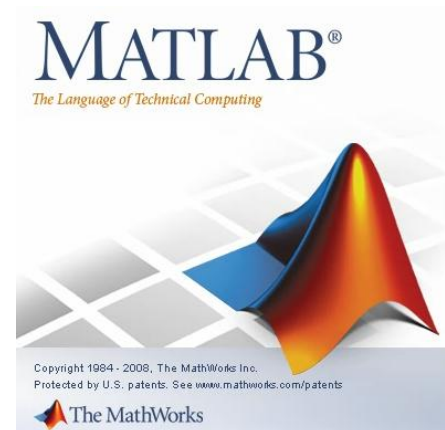


Abb. 2 - MR-Bild

Einführung in Matlab®

Wir haben für unser Projekt die Programmiersprache Matlab verwendet, die derzeit eine der wichtigsten technisch-mathematischen Programmiersprachen ist. Sie wurde von der Firma MathWorks, Inc. entwickelt und basiert zum Großteil auf Matrizen.

Matlab verwendet fast alle gängigen Programmiermethoden und besitzt auch unzählige vorgefertigte Funktionen.



Die wichtigsten Befehle/Funktionen:

➤ Variablen

Variablen müssen in Matlab nicht, wie in vielen anderen Programmiersprachen, deklariert werden, ihr Typ wird automatisch bestimmt.

➤ Matrizen

Matrizen sind ein- oder mehrdimensionale Datenfelder. In Matlab greifen viele Funktionen auf Matrizen zurück.

➤ Schleifen

Als Schleife wird ein Prozess bezeichnet, der sich beliebig oft wiederholt.

Die wichtigsten Schleifen sind die for-Schleife, in der die Variable automatisch in jedem Schleifendurchlauf erhöht wird, und die while-Schleife.

➤ if-Abfrage

Die Befehle innerhalb eines Anweisungsblocks werden nur ausgeführt wenn das Statement in der Abfrage wahr ist.

➤ Zeichenfunktionen

„plot“ zur Darstellung von eindimensionalen Matrizen als Graphen

„surf“ zur dreidimensionalen Darstellung von zweidimensionalen Matrizen

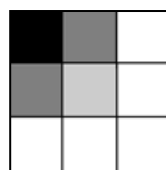
Handhabung von Bildern in Matlab:

Bilder werden in Matlab als zweidimensionale Matrizen interpretiert. Die Werte der Matrix stellen die Intensitäten des Bildes dar (0 = schwarz, 255 bzw. 1 = weiß). Daher können alle für Matrizen bestimmten Funktionen auch auf Bilder angewandt werden.

BEISPIEL:

0	0.5	1
0.5	0.8	1
1	1	1

Matrix



Bild

Konturen-Vektor-Segmentierung



Abb. 1: Phantom

Mit diesem Verfahren können Bilder mit einfachen, runden Formen und glatten Kanten, wie das „Phantom“ (Abb.1), optimal segmentiert werden.

Der erste Schritt besteht darin, die Konturen, also die Grenzen zwischen den verschiedenen Intensitäten des Bildes (in der Abbildung die verschiedenen Graustufen), zu finden. Dazu werden zunächst alle verschiedenen Intensitäten des Bildes gesucht und dann für jede Intensität folgende Schritte durchgeführt:

- Alle Bildpunkte dieser Intensität werden in ein neues Bild mit schwarzem Hintergrund kopiert und weiß eingefärbt.
- Nun wird ein drittes Bild erstellt, in dem alle Bildpunkte des zweiten Bildes, die entweder schwarz sind (d.h. außerhalb des weißen Bereichs liegen) oder weiß sind und mindestens 6 weiße Nachbarbildpunkte haben (d.h. innerhalb des weißen Bereichs liegen), schwarz bleiben. Nur jene weißen Pixel, die im zweiten Bild mehr als 2 schwarze Nachbarn hatten (d.h. am Rand des weißen Bereichs lagen), werden auch im dritten Bild weiß eingefärbt.
- Somit bleiben im dritten Bild nur die weißen Konturen einer Intensität mit schwarzem Hintergrund übrig.

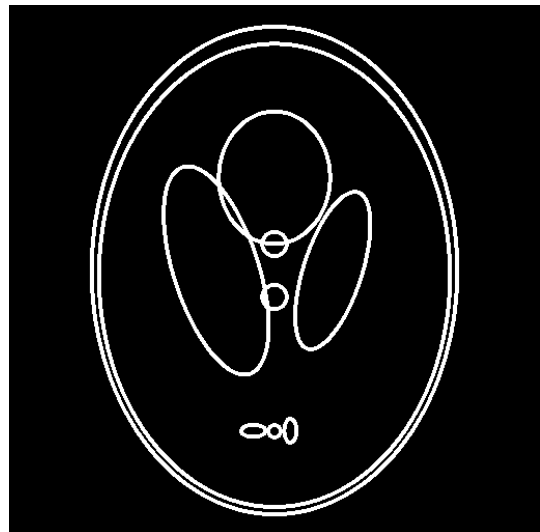


Abb. 3 - Phantom nach dem Konturen-Verfahren

Nachdem die Konturen aller Intensitäten wieder zusammengefügt werden entsteht ein schwarzes Bild, das nur die weißen Konturen des Originalbildes enthält (Abb. 2).

Im zweiten Schritt werden anhand der im ersten Schritt entstandenen Konturen die einzelnen Segmente ausfindig gemacht. Das Programm simuliert dabei das in der menschlichen Psyche verankerte „Gesetz der Einfachheit“, das uns im originalen Phantom Flächen als zusammengehörig erkennen lässt, die unterschiedliche Intensitäten haben.

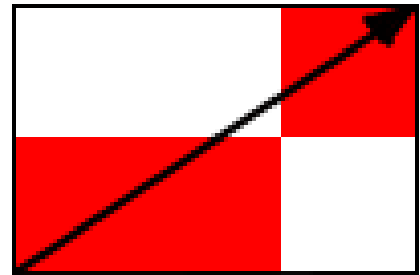
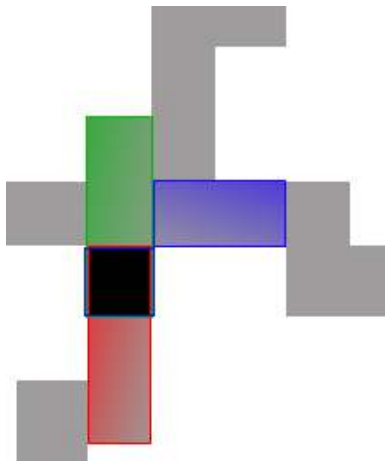


Abb. 3 - Der Vektor (3|2) als Matrix:

0	0	1
1	1	0

Daher werden sie vom Computer als verschiedene Segmente gedeutet, die uns aber durch ihre einfache, logische Form als ein Objekt erscheinen. Das Verfahren basiert auf Vektoren, die in Form von kleinen Matrizen mit dem Bild, das ebenfalls als Matrix vorhanden ist, überlagert werden. Nach mehreren Fehlversuchen hat sich als Größe der Matrizen je nach Richtung 1x3, 2x3 bzw. 3x2 und 3x3 als günstig erwiesen (Abb.3).



Das Verfahren arbeitet folgendermaßen: Von einem aus dem Bild, das die Konturen enthält, zufällig gewählten weißen Punkt werden in 24 Richtungen Vektoren gebildet. Aus allen Vektoren, die wiederum auf einen weißen Punkt zeigen, wird einer zufällig ausgewählt. Die drei Pixel, die auf diesem Vektor liegen werden in einem neuen Bild weiß eingefärbt. Von dem Punkt, auf den der Vektor zeigt, werden wieder 24 Vektoren gebildet. Jetzt aber werden alle Vektoren, die auf einen weißen Punkt zeigen, mit dem vorherigen Vektor verglichen und der ähnlichste wird ausgewählt.

Abb. 4 - An diesem Kreuzungspunkt würde die grüne Vektormatrix der blauen bevorzugt, da sie der vorherigen (roten) ähnlicher ist.

So entsteht eine Kettenreaktion, die erst wieder zum Stillstand kommt, wenn ein Vektor wieder den Ausgangspunkt erreicht. An Kreuzungspunkten der Konturen verschiedener Segmente wählt das Programm durch dieses Verfahren den Vektor aus, der die gerade Linie am besten fortsetzt und dadurch werden auch sich überlappende Segmente als solche erkannt (Abb.4).

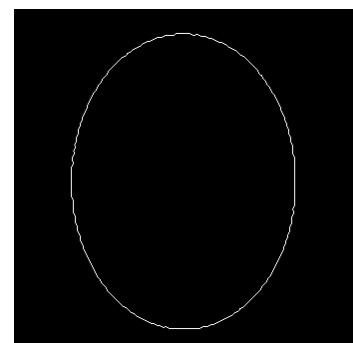
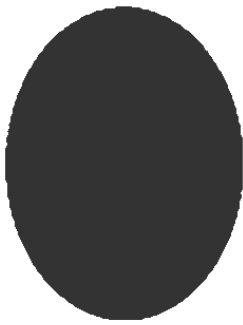


Abb. 5 - Kontur eines Segments

Das neue Bild, das sich aus den weiß gefärbten Bildpunkten aller Vektoren zusammensetzt, enthält dann die Kontur eines einzelnen Segments (Abb.5).



Als letzter Schritt wird jedes Segment in einer der Farben bzw. Intensitäten, die es im originalen Bild enthalten hat, eingefärbt, der Hintergrund je nach Segmentfarbe schwarz oder weiß. Das Ergebnis sind einige Bilder, von denen jedes ein Segment in der richtigen Farbe darstellt (Abb.6).

Abb. 6 - Segment, in der richtigen Farbe dargestellt

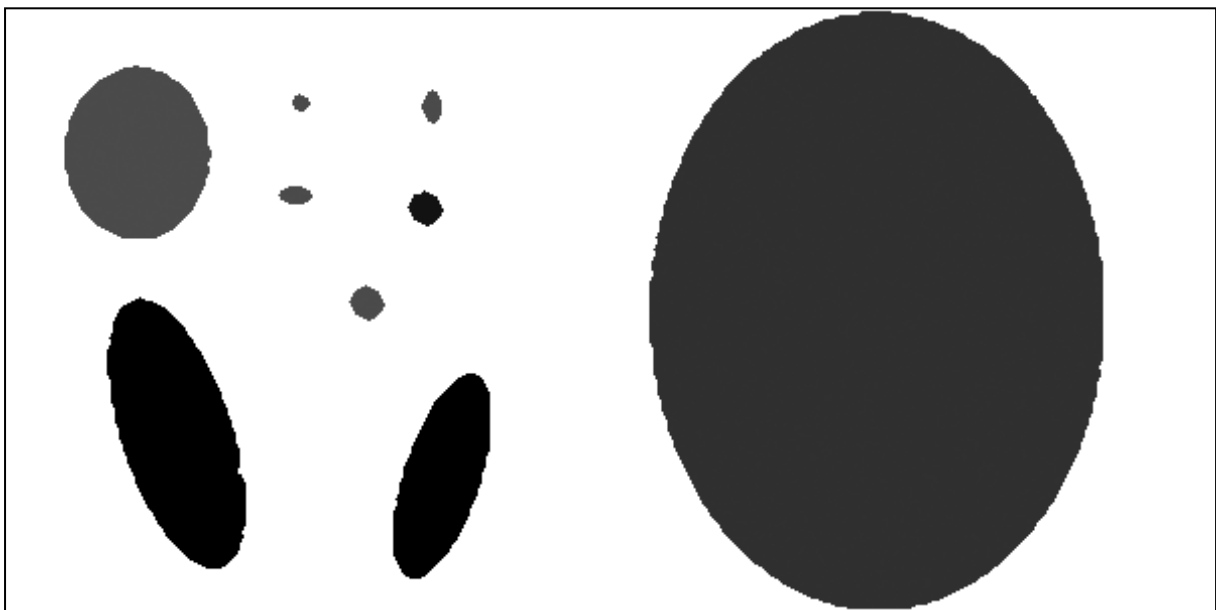


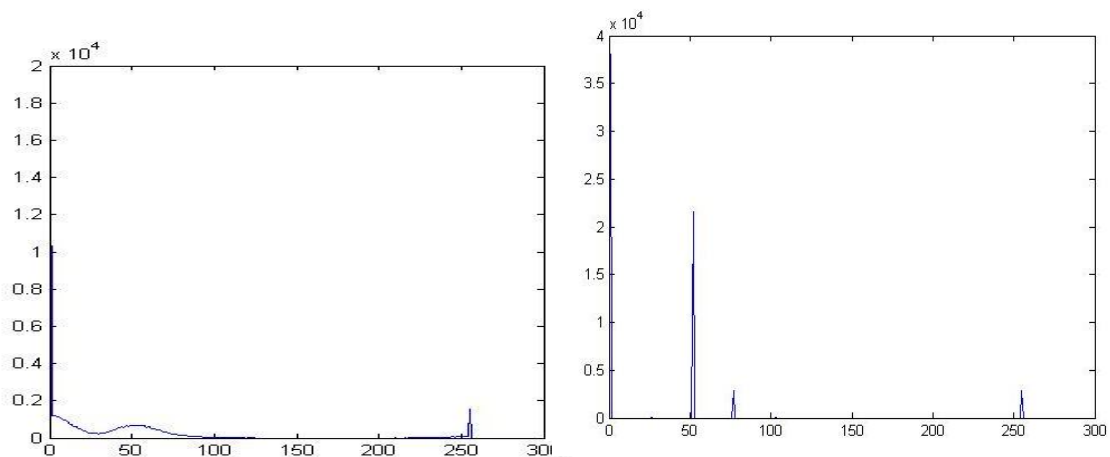
Abb. 7 - Alle Segmente des Phantoms

Allgemeine Segmentierung

Rauschen

Histogramm

Um die Grenzen der Segmente im verrauschten Phantombild möglichst gut wählen zu können, stellten wir zu Beginn ein Histogramm des Bildes auf, in welchem auf der x-Achse die Intensität, und auf der y-Achse aufgetragen ist, wie viele Pixel mit der zugehörigen Intensität im Bild enthalten sind. Da Rauschen im Allgemeinen meistens einem zufälligen Faktor mit Normalverteilung entspricht, erhofften wir vom Histogramm, auf das ursprüngliche Bild schließen zu können. Die folgenden Grafiken zeigen das Histogramm des Phantombildes, mit und ohne Rauschen:

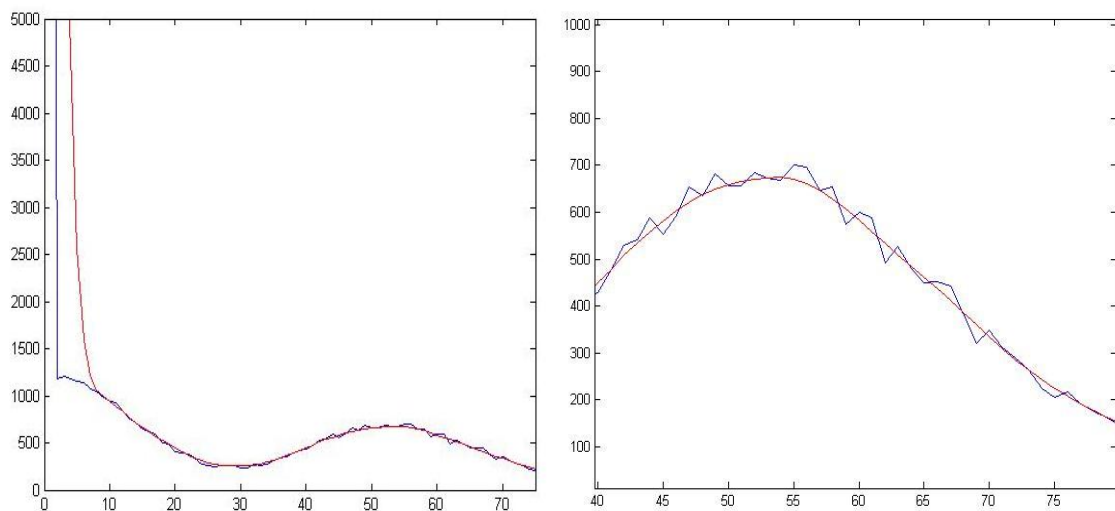


Man erkennt leicht, dass der Pik an der Stelle

51 auf eine Gauß'sche Glockenkurve aufgeteilt hat, doch der Pik an der Stelle 76 geht darin unter. Um mit den Werten besser arbeiten zu können, glätteten wir die Kurve nach der Formel:

$$X_i = \frac{X_{i-1} + 2X_i + X_{i+1}}{4}$$

Danach entspricht jeder Wert dem Durchschnitt der angrenzenden Werte und sich selbst (doppelt gewichtet). Nach mehrmaligem Anwenden resultierte folgendes Ergebnis:



Unser nächstes Ziel war es die Tiefpunkte herauszufinden und damit später auf die Höhepunkte der Gauß'schen Glockenkurven zu kommen. Mit den Höhepunkten wäre eine Rekonstruktion der Grauintensitäten vor der „Verunreinigung“ durch das Rauschen durchführbar. Die Tiefpunkte Das Problem stellten unter anderem der untergegangene Pik an der Stelle 76 dar sowie auch

Das setzten wir mit einem dynamischen System um, wobei P den Vektor der Grenzen zwischen den einzelnen Segmenten darstellte.

$P'_i = -2P_i + P_{i-1} + P_{i+1} - f'(P_i)$, wobei f die Funktion des Histogramms ist. Aber dieser Weg führte zu keinem Ergebnis, weil sich das System meistens aufschaukelte und nicht konvergierte.

Varianz-Optimierte Segmentierung:

Zur Segmentierung des MR-Bildes fassten wir Intervalle von Intensitäten zu Teilmengen zusammen. Eine gleichmäßig verteilte Grenzwahl brachte nicht die gewünschten Ergebnisse, da bestimmte Intensitäten seltener vorkamen als andere und oft auch nur sehr verstreut und nicht kompakt im Bild vorhanden sind. Darum verwendeten wir eine Methode zur Minimierung der Varianz zwischen dem Modell und dem Originalbild. Bei dieser Vorgangsweise wird das Minimum folgender Funktion gesucht:

$$\| a_1 * A_1 + a_2 * A_2 + \dots + a_n * A_n - I \|^2$$

Ableiten nach d/da_i ergibt:

$$a_i = \frac{\sum_{x,y} (I_{x,y} * A_{i,x,y})}{\sum_{x,y} A_{i,x,y}}$$

Der rechte Ausdruck entspricht dem Durchschnitt aller Intensitäts-Werte des Originalbildes an den Stellen, an denen $A_i = 1$ gilt, denn der Nenner des Bruches gibt die Anzahl der Pixel wider, während der Zähler die Intensitäten aufsummiert. Wenn also die A_i fixiert sind, können die a_i mit dem Mittelwert berechnet werden. Um die A_i bei gegebenen a_i zu berechnen, haben wir die Mittelwerte zwischen zweier aufeinander folgender a_i bestimmt und jenen Bereich als neues Segment für A_i verwendet, in dem die Pixel des Original-Bildes zwischen zweier dieser Mittelwerte liegen.

Wenn man mit dieser Methode alternierend die Bereiche der A_i und die Werte der a_i berechnet, so erhält man ein Modell, das nach endlichen Schritten konvergiert und sich nicht mehr verändert. Nach diesem Endergebnis legten wir unsere Segmente aus und erhielten folgende Einteilung für $n = 4$:



Dieses Segmentierungsverfahren hat das MR-Bild zwar sehr vereinfacht, doch es wurden noch nicht die einzelnen Organe als zusammengehörige Objekte erkannt. Um dies zu bewerkstelligen, änderten wir die obige Methode etwas ab, indem wir für die a_i eine quadratische Funktion einfügten, die auch ortsabhängig bezüglich der x- und y-Koordinaten des jeweiligen Pixels ist. Die Funktion sieht danach so aus:

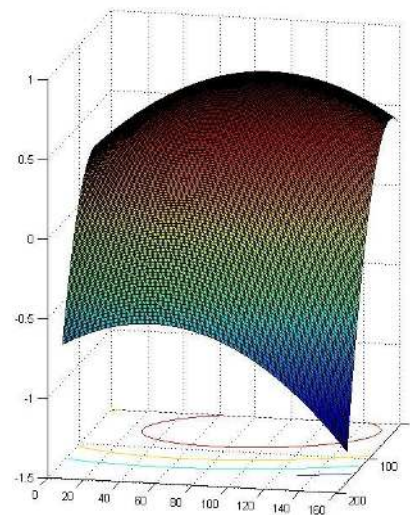
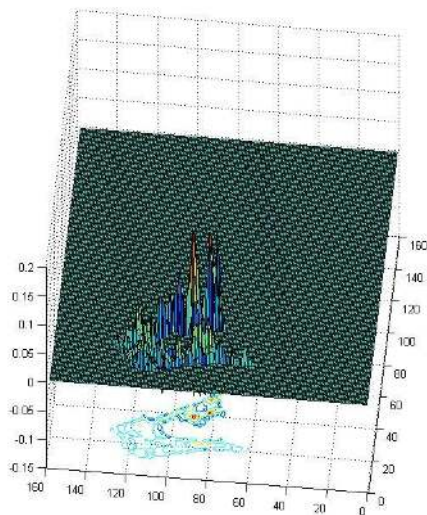
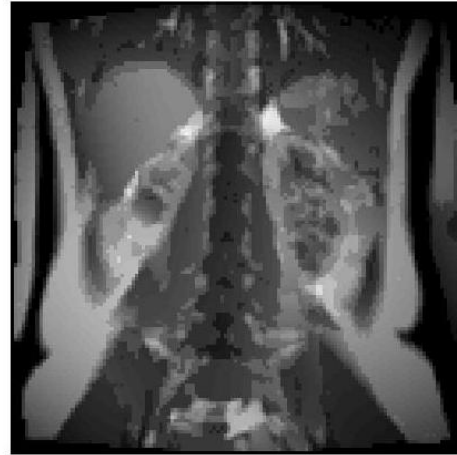
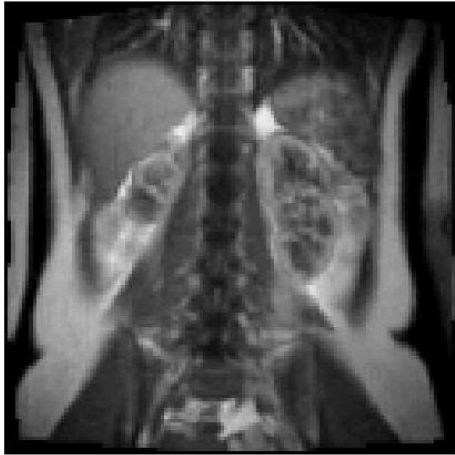
$$\| \sum_{i=1}^n (a_i + b_i * x + c_i * y + d_i * x * y + e_i * x^2 + f_i * y^2) * A_i - I \| ^2$$

Ableiten nach $\{a_i, b_i, \dots, f_i\}$ ergibt:

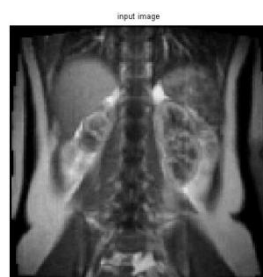
$$\begin{bmatrix} \sum A_i & \sum A_i x & \sum A_i y & \sum A_i xy & \sum A_i x^2 & \sum A_i y^2 \\ \sum A_i x & \sum A_i x^2 & \sum A_i xy & \sum A_i x^2 y & \sum A_i x^3 & \sum A_i xy^2 \\ \sum A_i y & \sum A_i xy & \sum A_i y^2 & \sum A_i xy^2 & \sum A_i x^2 y & \sum A_i y^3 \\ \sum A_i xy & \sum A_i x^2 y & \sum A_i xy^2 & \sum A_i x^2 y^2 & \sum A_i x^3 y & \sum A_i xy^3 \\ \sum A_i x^2 & \sum A_i x^3 & \sum A_i x^2 y & \sum A_i x^3 y & \sum A_i x^4 & \sum A_i x^2 y^2 \\ \sum A_i y^2 & \sum A_i xy^2 & \sum A_i y^3 & \sum A_i xy^3 & \sum A_i x^2 y^2 & \sum A_i y^4 \end{bmatrix} \begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \\ e_i \\ f_i \end{bmatrix} = \begin{bmatrix} \sum A_i I \\ \sum A_i Ix \\ \sum A_i Iy \\ \sum A_i Ixy \\ \sum A_i x^2 \\ \sum A_i y^2 \end{bmatrix}$$

Es wurde hierbei nur ein Segment A hergenommen, für das ein Quadratisches Modell erstellt wird. Dieses Segment haben wir zuvor mit Hilfe der anderen Segmentier-Methoden erhalten. Durch Ableiten dieser Funktion nach $a, b, c, \dots, f$ erhält man sechs Gleichungssysteme.

Nachdem das quadratische Modell aufgestellt war, ließen wir das Segment wachsen, indem wir Pixel hinzufügten, die am Rand des Segmentes lagen und eine Intensität hatten, die nicht weiter als ein bestimmter Toleranzbereich vom berechneten Wert dieser Stelle abwichen. Wir mussten den Toleranzbereich variieren, bis wir schließlich auf folgendes Endergebnis kamen:



Es ist unverkennbar, dass die Leber und das Fettgewebe vollständig erkannt wurden. Das Quadratische Modell für das Fettgewebe sieht folgendermaßen aus:



Kontrastierung:

Bei der Kontrastierung geht es im Allgemeinen darum, die einzelnen Teile eines Bildes hervorzuheben, indem man die Intensitäten des Bildes zusammenfasst und somit einen härteren Farbverlauf erzielt.

Nachdem das Bild eingelesen und einer Variablen zugewiesen wurde ist es wichtig, zu wissen, wie viele verschiedene Intensitäten im Bild vorhanden sind.

Dieser Schritt ist in Matlab sehr einfach, da es dafür eine vorgefertigte Funktion gibt: „unique(Name der Bildmatrix)“.

Diese Funktion gibt eine eindimensionale Matrix zurück, in der alle Intensitäten, der Größe nach geordnet, enthalten sind (Abb. 1).

Dieses Array ist die Basis für einen weiterführenden Algorithmus, der die Farben des Ausgangsbilds (Abb. 2) zusammenfasst und in das fertige Bild (Abb. 3) einträgt.

0
0.01
0.03
...
1

Abb. 1 –
Intensitäten-Matrix



Abb. 2 - Vor der Kontrastierung

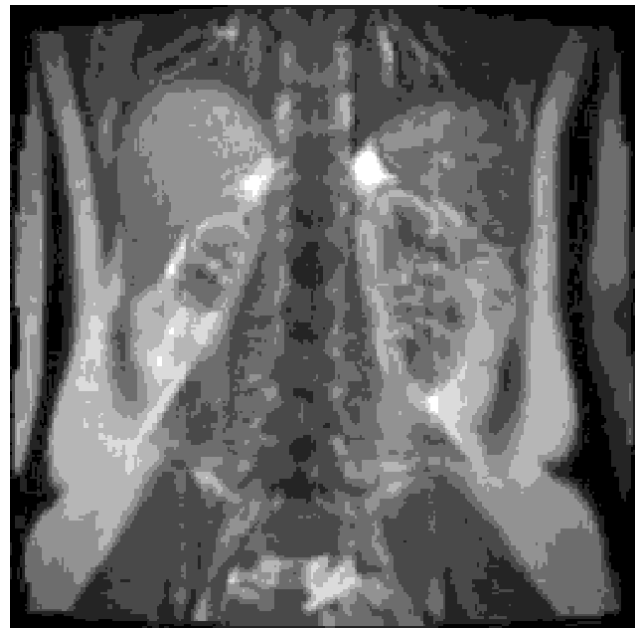


Abb. 3 - Nach der Kontrastierung

Glättung:

Bei der Glättung geht es im Allgemeinen darum, Kanten, die durch Kontrastierung oder Segmentierung eckig geworden sind, wieder zu glätten, ohne dabei relevante Informationen zu verlieren.

Erster Ansatz:

Im ersten Ansatz sind Regeln (Abb. 1) aufgestellt worden um Ecken zu erkennen und anschließend zu Glätten.

Jeder Pixel wird mit Hilfe einer Zeilen- und einer Spaltenschleife, die verschachtelt sind, auf jedes dieser Muster überprüft und falls ein Muster zutrifft, wird der entsprechende Pixel eingefärbt.

Diese Methode ist aber leider nicht ganz zufriedenstellend, weil sie dazu neigt, gerade Linien zu zeichnen, wodurch wichtige Informationen verloren gehen könnten.

Nach zu vielen Durchgängen wird das Bild bis zur Unkenntlichkeit geglättet, sodass nur noch ein graues Bild übrig bleibt.

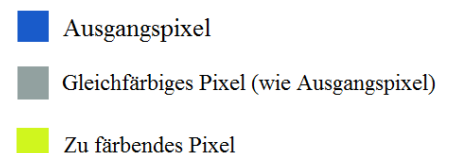
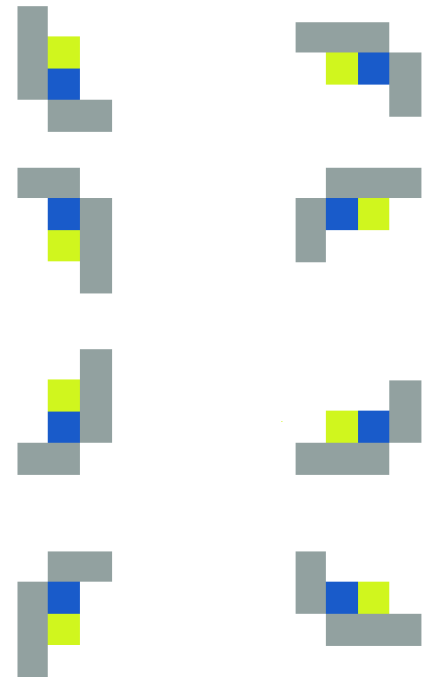


Abb. 1 - Alle Regeln, grafisch dargestellt

Zweiter Ansatz:

Der zweite Ansatz zur Glättung ist im Eigentlichen eine Art Mittelungsprozess.

Hierbei kommen wieder die Zeilen- und die Spaltenschleife zum Einsatz.

Bei jedem Pixel wird das direkte Umfeld überprüft, die verschiedenen Werte in eine Ungleichung eingetragen und dann der Wert in der Mitte in den Ausgangspixel eingetragen.

Diesen Prozess erledigt praktischerweise die Matlab-Funktion „median“.

Nach mehreren Durchgängen ergibt sich so eine Glättung.

BEISPIEL:

P... Ausgangspixel (in diesem Fall hat es den Wert 0)

Überprüfungsbereich:

0.6	1	1
1	P(0)	0
1	1	1

Median teilt es auf in: $1 > \underline{0.6} > 0$

Der Wert 0.6 liegt in der Mitte, deswegen wird er in den Pixel P eingetragen.

(Noch korrekter wäre es, statt „median“ den Befehl „mode“ zu verwenden, welcher aber viel mehr Ressourcen braucht und daher den Prozess unproportional zur Besserung des Bildes verlangsamt.)

Der Vorteil gegenüber dem ersten Ansatz ist, dass das Bild auch nach vielen Durchläufen nicht zur Bildung von Linien tendiert und man den Durchlaufparameter beliebig erhöhen kann, ohne dass das Bild sich verzerrt.

Wichtig: Auf Details achten!



Abb. 1 - Vor der Glättung

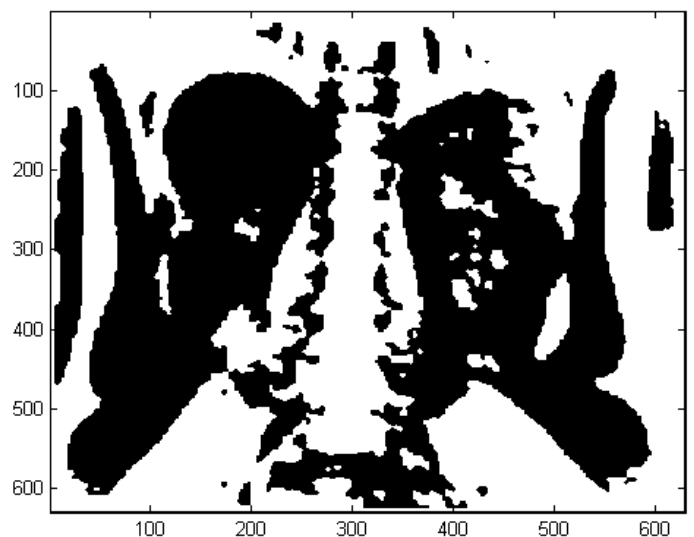


Abb. 2 - Nach der Glättung

Interaktive Segmentierung

Ein Teilziel der Gruppe war es, ein interaktives Programm zu erstellen, das es beispielsweise einem Arzt ermöglicht, durch manuelle Einstellungen einen Teil oder ein Organ eines MR-Bildes hervorzuheben und eine optimale Segmentierung zu erstellen.

Für dieses Programm wurden Codes aus Matlab in ein Unterprogramm namens Simulink übertragen, in dem es möglich ist durch grafische Blöcke wie Sliders (Regler) Variablen zu verstellen und damit das Programm flexibel an jegliches Bild anzupassen.

Für die Arbeit wurde ein MR-Bild eines menschlichen Unterleibes verwendet.

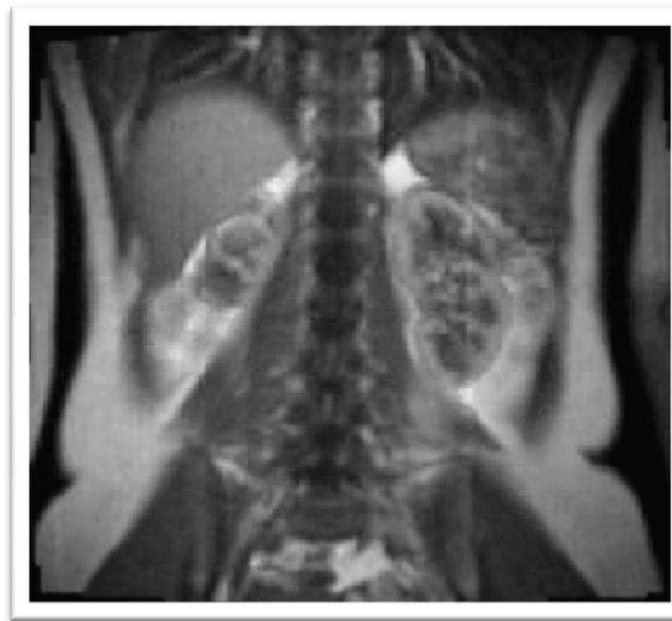


Abb. 1: Originalbild

Das dazu entwickelte Programm besteht aus fünf einfachen Schritten, die in weiterer Folge noch erklärt werden.

Durch Bearbeitung des Kontrasts, Zuschneiden des Bildes, Glätten des Ausschnitts, Einstellung von Schwellenwerten und Ausschneiden des Objekts mit Polygonen wird aus dem Bild ein einzelnes Segment herausgehoben.

Das hierfür verwendete Programm Simulink lässt sich über einen Button in der Taskleiste von Matlab aufrufen und wurde entwickelt, um die sehr theorielastigen Prozesse aus Matlab zu strukturieren und anschaulicher zu machen. Es ist, wie gesagt, auch möglich, es für interaktive Zwecke zu nutzen.

Das System von Simulink basiert auf unterschiedlichen Blöcken, mit verschiedenen Funktionen, die man auf der Benutzeroberfläche einfügen und beliebig miteinander kombinieren kann.

Kontrastieren:

Das Programm beginnt damit, den Kontrast des Bildes zu erhöhen, um eine stärkere Ausprägung der Konturen zu erhalten. Dies geschieht durch das Zusammenfassen der Intensitäten des Bildes.

Jeder einzelne Bildpunkt auf dem Bild hat eine Intensität zwischen 0 und 255, welche durch das Dividieren durch das Maximum zu Intensitäten zwischen 0 und 1 ummodelliert werden (0 = schwarz, 1 = weiß).

Das Zusammenfassen geschieht folgendermaßen: Alle Intensitäten, die kleiner als 0.2 sind, werden auf 0 reduziert werden, alle, die kleiner als 0.4 sind, werden auf 0.2 reduziert usw. In Matlab wird dieser Vorgang durch eine for- Schleife vereinfacht.

Der Benutzer kann mit den Slidern einstellen, wie viele Intensitäten jeweils zusammengefasst werden und das Bild individuell an die eigenen Bedürfnisse anpassen.

Direkt im Code befindet sich der Befehl, das Ergebnis zu speichern, wodurch man nahtlos zum nächsten Schritt weitergehen kann, welcher sofort das Ergebnis des vorherigen Codes aufruft.

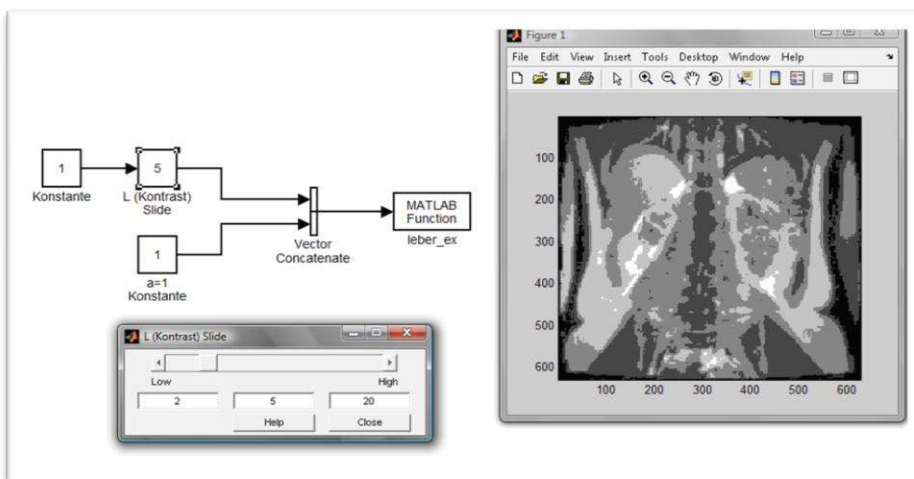


Abb. 2: Vergleichsobjekt Kontrast 1

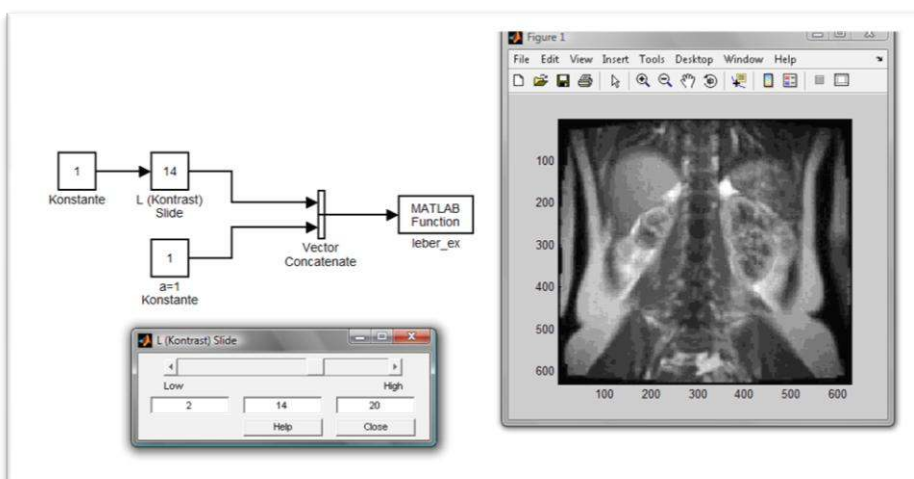
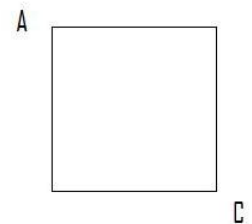


Abb. 3: Vergleichsobjekt Kontrast 2

Zuschneiden:

Der nächste Schritt ist es, das Bild auf die Größe und Form des zu segmentierenden Elements zuzuschneiden.

Hier wird das Feld mithilfe von 2 variablen Koordinaten verkleinert, die sich wieder im Simulink verändern lassen. Der Code benutzt die beiden Koordinaten als Eckpunkte (A, C) für ein Rechteck, aus dem das neue Bild entsteht.



Der Code beinhaltet ebenfalls eine Sperre, die verhindert, dass C kleiner als A ist, da sonst die Formel nicht mehr funktionieren würde.

Abb. 4: Schema

Natürlich muss immer auf die Größe des ursprünglichen Bildes geachtet werden, damit man keine zu kleinen Bereiche ausschneidet oder die Koordinaten außerhalb des Bildes setzt.

In diesem Beispiel wird der Fokus auf die Leber gesetzt.

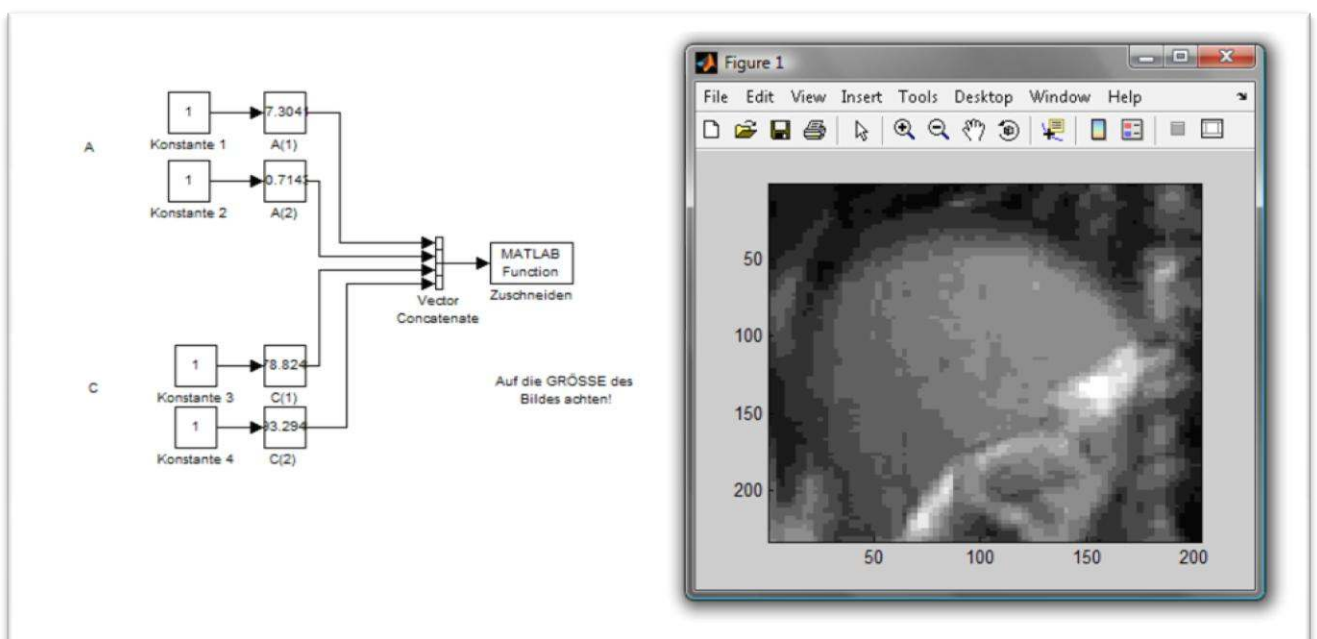


Abb. 4: Zugeschnittenes Bild der Leber

Glättung:

Für die folgende Aktion wird der bereits beschriebene Code der Glättung nach Pixeln verwendet.
(→siehe Beschreibung der Glättung)

Der einzige Unterschied besteht darin, dass der Code für das interaktive Verfahren wieder auf Simulink übertragen wurde und der Glättungsfaktor als Variable mit Sliders geregelt werden kann.
Je höher dieser Faktor ist, desto besser wird das Bild geglättet.

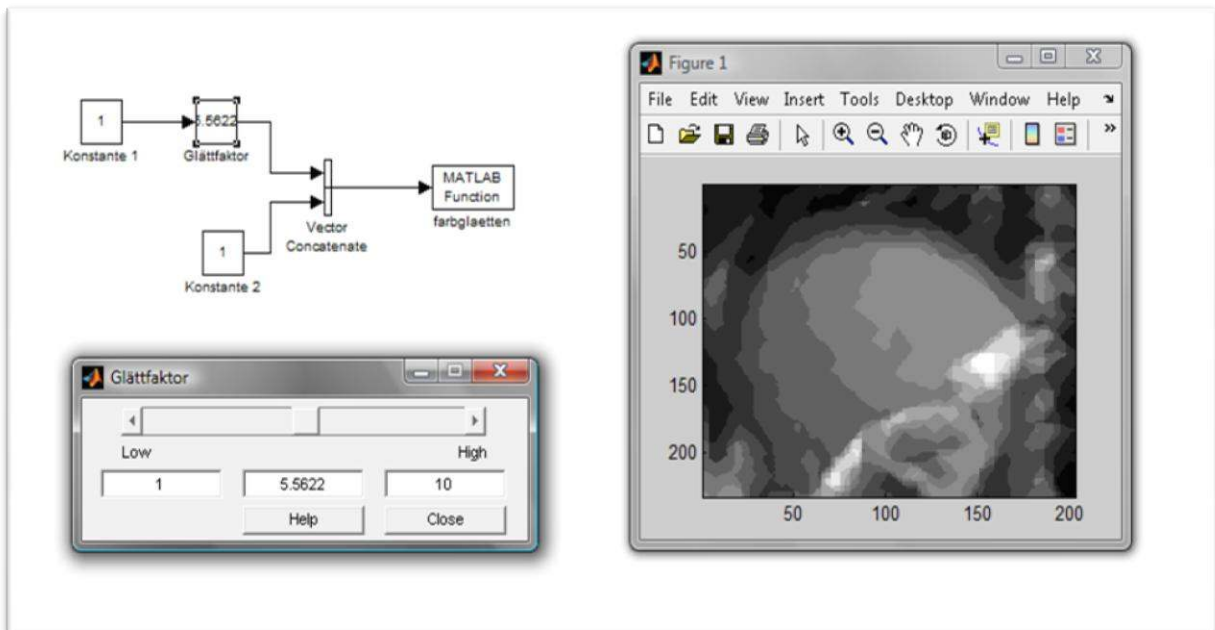


Abb. 5: Glättung des Ausschnitts

Schwellenwerte:

Um ein besseres Bild von der Form der Leber zu erhalten, werden zwei „Schwellenwerte“(a, b) festgesetzt, alle Bildpunkte, deren Intensität zwischen a und b liegt, werden weiß und alle Punkte, die eine Intensität außerhalb von a und b haben, schwarz eingefärbt.

Und wieder werden die Schwellenwerte in Simulink zu Variablen gemacht und können manuell adjustiert werden.

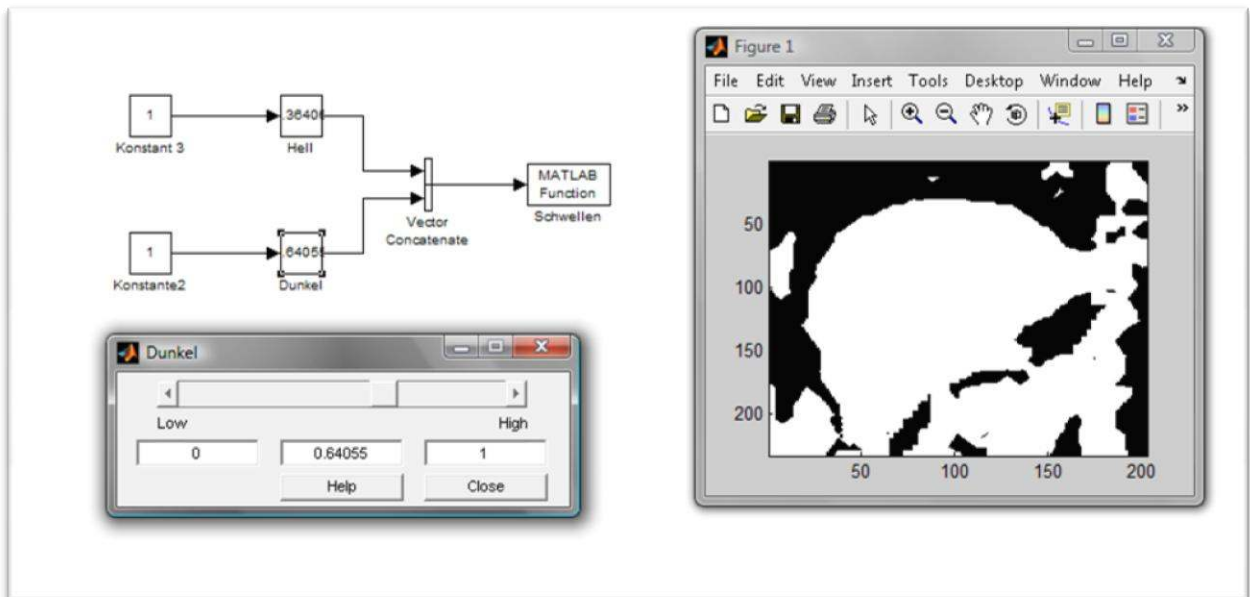


Abb. 6: Einstellung der Schwellenwerte

Ausschneiden:

Zu guter Letzt müssen aus dem nun binären Bild noch die überschüssigen Teile, die nicht zur Leber gehören, aber aufgrund der gleichen Intensitäten in den vorherigen Schritten noch nicht eliminiert werden konnten, entfernt werden.

Dies geschieht durch das platzieren von vier Polygonen, die zwar durch Konstanten an die Ecken des Bildes gebunden sind, aber dennoch frei bewegt werden können.

Leider war es nicht möglich, alle vier Gleichungen in eine Funktion zusammenzufassen, wodurch man für diesen Schritt vier beinahe identische Simulink-Anwendungen ausführen muss.

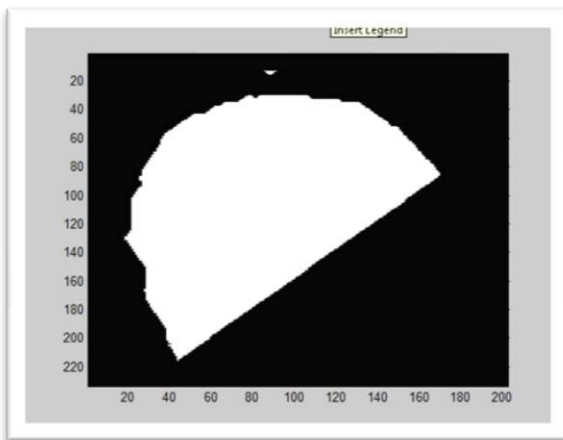
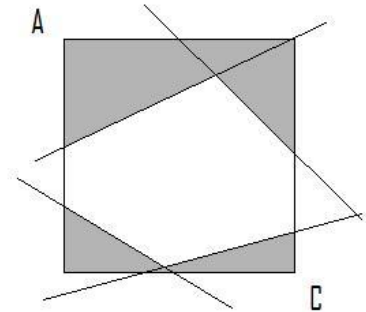


Abb. 7 Fertigter Ausschnitt der Leber

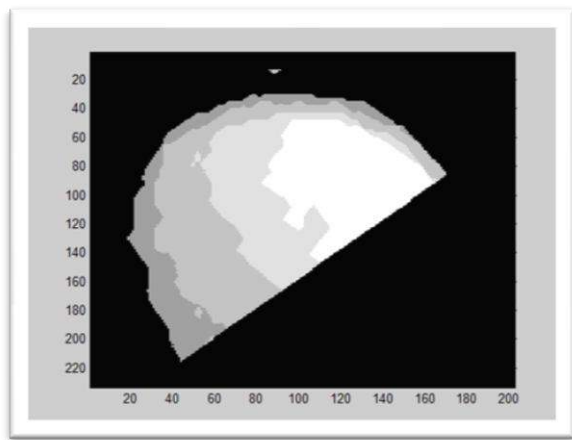


Abb. 8: Segment der Leber

Es werden Geraden im Koordinatensystem konstruiert. Für alle Werte zwischen der Geraden und dem Ursprung gilt $y < kx + d$, wohingegen $y > kx + d$ für die andere Richtung gilt.

Je nachdem, an welche Ecke das Polygon gebunden wird, wird der äußere Teil ($y >$ oder $y <$) schwarz gefärbt.

Als letzter Schritt kann das gewünschte Objekt noch mithilfe einer Matlab-Funktion als fertiges Segment angezeigt und gespeichert werden (Abb.9).