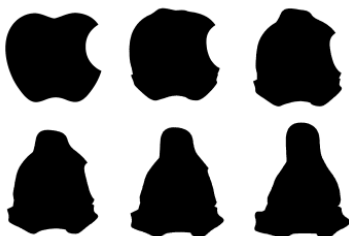
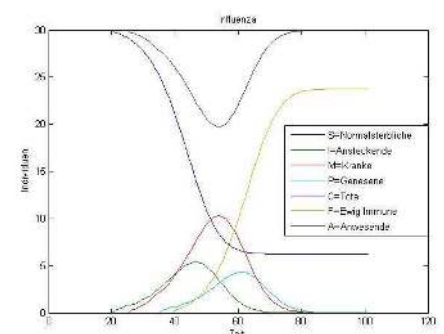
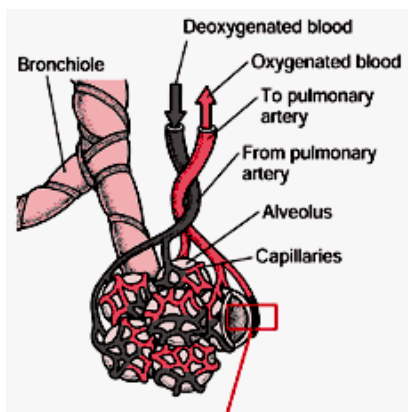


WOCHE DER MODELLIERUNG MIT MATHEMATIK

Dokumentationsbroschüre

7. - 13. Jänner 2006



Das Land
Steiermark

Bank Austria
Creditanstalt



ZUKUNFTsfonds
STEIERMARK



Woche der Modellierung mit Mathematik



Schloss Seggau, 07.01. - 13.01.2006

Weitere Informationen:

<http://math.uni-graz.at/modellwoche/2006/>

Vorwort

Die Idee zu der in der Steiermark durchgeführten „Modellierungswoche“ für Schüler der 7. und 8. Klasse der AHS wurde schon längere Zeit am Institut für Mathematik und Wissenschaftliches Rechnen der Universität Graz diskutiert. Vorbild waren ähnliche Vorhaben, die bereits in Kaiserslautern, in Bozen und erstmals im letzten Jahr auch in Linz durchgeführt wurden. Mitglieder des Institutes haben bereits Erfahrungen mit ähnlichen Veranstaltungen für Studierende und angehende Wissenschaftler. Im Jahre 2005 wurde vom Institut für Mathematik und Wissenschaftliches Rechnen der Karl-Franzens-Universität erstmals eine Modellierungswoche durchgeführt, die bei allen teilnehmenden SchülerInnen großen Anklang gefunden hat. Ermutigt durch diesen durchschlagenden Erfolg haben wir auch im Jahre 2006 wieder eine Modellierungswoche angeboten. Hauptziel der Modellierungswoche war es, Schüler mit einem Aspekt der Mathematik zu befassen, der unserer Meinung nach im Unterricht an den AHS unterrepräsentiert ist: Die Rolle der Mathematik als Werkzeug zum Verständnis der Welt, die uns in Alltag und Wissenschaft umgibt. Während ihrer gesamten Geschichte stand die Mathematik immer in Wechselwirkung mit angewandten Bereichen. Viele mathematische Theorien entstanden in Reaktion auf Anforderungen aus den verschiedensten Anwendungsbereichen. Die Verfügbarkeit immer leistungsfähigerer Computer hat neue Möglichkeiten für die mathematische Behandlung verschiedenster komplexer Probleme eröffnet. Quantitative Resultate statt qualitativer Aussagen sind immer wichtiger und erfordern zu ihrer Bewältigung die mathematische Modellierung komplexer Systeme in interdisziplinärer Zusammenarbeit.

Den an der Modellierungswoche teilnehmenden SchülerInnen sollte an Hand sorgfältig ausgewählter Projektaufgaben Gelegenheit gegeben werden, den angewandten Aspekt der Mathematik durch Teamarbeit in Projektgruppen zu erleben. Es wurde versucht, den Teilnehmenden die wesentlichen Phasen eines Modellierungsprozesses nahe zu bringen: Einarbeiten in das Anwendungsgebiet, Wahl der Modellstruktur in Hinblick auf die Aufgabenstellung, Einsatz numerischer Methoden, Interpretation der Ergebnisse, Präsentation der Resultate.

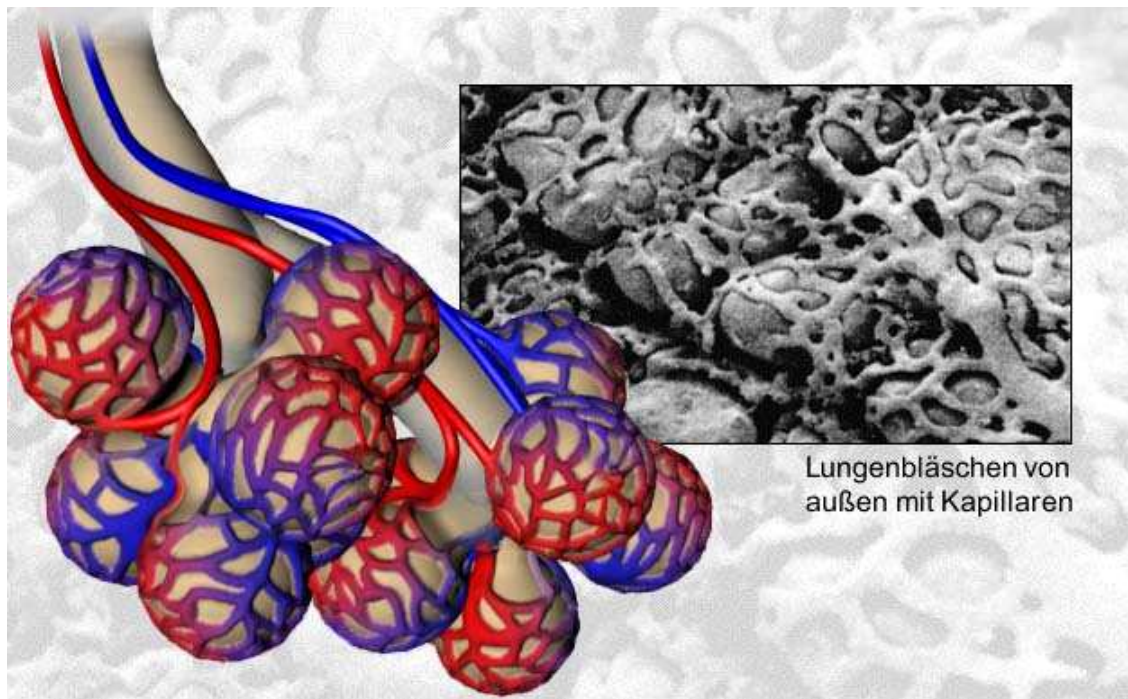
Treibende Kraft für die Realisierung der Modellierungswoche war Dr. Stephen Keeling, dem hier für seinen großen Einsatz gedankt sei. Besonderer Dank gebührt dem Landesschulrat für Steiermark, und hier insbesondere Frau Landesschulinspektor Hofrat Mag. Marlies Liebscher. Sie hat die Idee einer gemeinsamen Veranstaltung sofort sehr positiv aufgenommen und tatkräftig unterstützt. Ohne den großen Einsatz der direkten Projektbetreuer, Dr. Sigrid Thaller – Institut für Sportwissenschaft, Dr. Wolfgang Desch, Dr. Stephen Keeling, Dr. Georg Propst – alle Institut für Mathematik und Wissenschaftliches Rechnen, und der Betreuer aus dem Kreis der Lehrerschaft, Mag. Theresa Gradnitzer, Mag. Erna Hammerl und Mag. Melanie Wogrin, die auch die Gestaltung dieses Berichtes übernommen haben, sowie Dr. Alfio Borzì vom Institut für Mathematik und Wissenschaftliches Rechnen wäre die Modellierungswoche nicht durchführbar gewesen. Der Bank Austria – Creditanstalt sei für eine nicht unbeträchtliche Subvention gedankt. Dem Programm für Begabten- und Begabungsförderung Steiermark sei für eine wesentliche Subvention gedankt. Für seine Unterstützung sei Vizerektor Polaschek, Universität Graz, gedankt.

Schloss Seggau, am 13. 1. 2006

F. Kappel
(Leiter des Institutes für Mathematik und
Wissenschaftliches Rechnen)



Gasaustausch in der Lunge



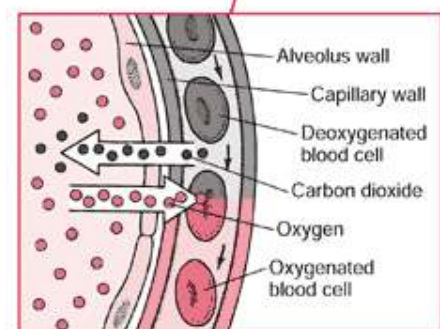
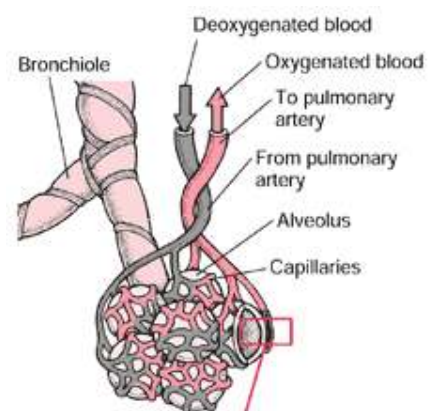
Vorstellung der Fragestellung

Wir beschäftigen uns mit der Frage, wann der Gasaustausch in einer menschlichen Lunge maximal wird, wovon dieser abhängig ist und welche mathematischen und physikalischen Prinzipien diesem zugrunde liegen. Mit den erhaltenen Erkenntnissen haben wir ein mathematisches Modell erstellt.

Einleitung

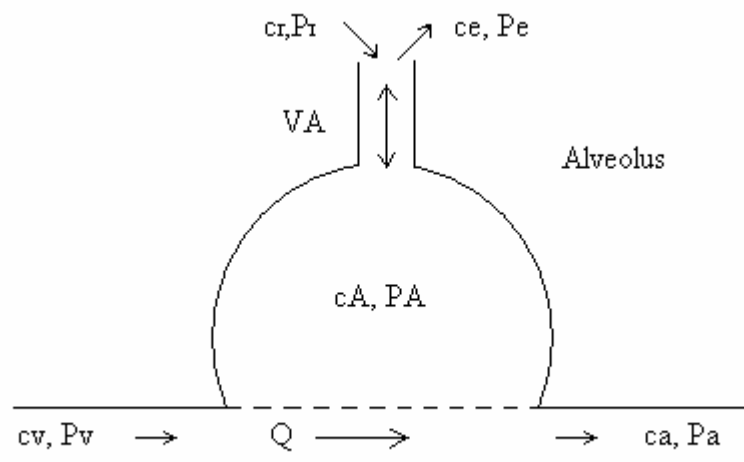
Die menschliche Lunge enthält ca. 300 Mio. kleine Bläschen, die so genannten Alveoli. In diesen Alveolen findet der Kontakt zwischen Blut und Atemluft in suffizientem Maße statt, sodass eine ausreichende Versorgung mit Sauerstoff und der Abtransport von diversen Gasen wie z.B. CO_2 aus dem Blut gewährleistet werden kann. Eine semipermeable Membran bildet die Grenzfläche zwischen Atemluft und Blut, welche durch ihre sehr dünne Beschaffenheit die Diffusion sehr begünstigt. Aufgrund dieser physiologischen Gegebenheiten können wir annehmen, dass sich der Gehalt an Gas in der Atemluft und im Blut im Gleichgewicht befindet.

Das Blut wird von der rechten Herzkammer über die Arteria pulmonalis in die vielfach verzweigten Lungenkapillaren gepumpt und erreicht auf diesem Weg schließlich die Alveolen – so ist die Perfusion (Durchblutung) dieser gegeben. Hat der Gasaustausch stattgefunden,



sammelt sich das Blut in der Vena pulmonalis und erreicht so den linken Vorhof und schließlich die linke Kammer von wo es seinen Weg durch den gesamten Kreislauf nimmt. Es ist anzunehmen, dass die Summe der Blutflüsse durch die Alveolen dem gesamten Blutfluss der Lunge entspricht.

Die Versorgung der Lunge mit Atemluft wird als Ventilation bezeichnet und erfolgt über die Trachea, welche sich in die Bronchien aufspaltet, über die die Atemluft schließlich die Alveolen erreicht. Die Ausatmung erfolgt auf demselben Weg.



Parameter:

P	Partialdruck des betrachteten Gases
V	vom Gas eingenommenes Volumen
N	Anzahl der Gasmoleküle
K	Boltzmann Konstante
T	absolute Temperatur
Q	Blutfluss durch den Alveolus
C	Konzentration des betrachteten Gases
B	Löslichkeit des betrachteten Gases
I	eingeatmete Luft
A	alveolare Luft
E	ausgeatmete Luft
V	venöses Blut
A	arterielles Blut

Ideales Gasgesetz

$$PV = nkT$$

Partialdruck in einem Gasgemisch:

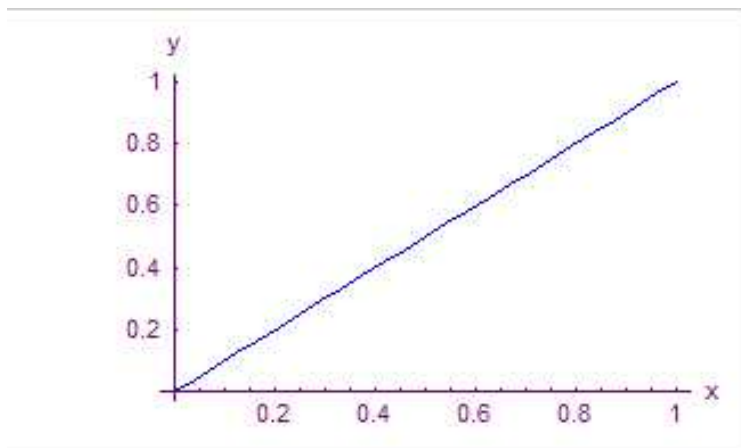
$$P = ckT$$

In einem Gasgemisch hat jedes Gas einen eigenen Partialdruck. Der Gesamtdruck des Gasgemisches ist die Summe der einzelnen Partialdrucke.

Partialdruck in einem gelösten Gas:

$$P = \frac{c}{\sigma}$$

Dieses Gesetz gilt nur für **einfache Lösungen** und ist eine lineare Funktion.



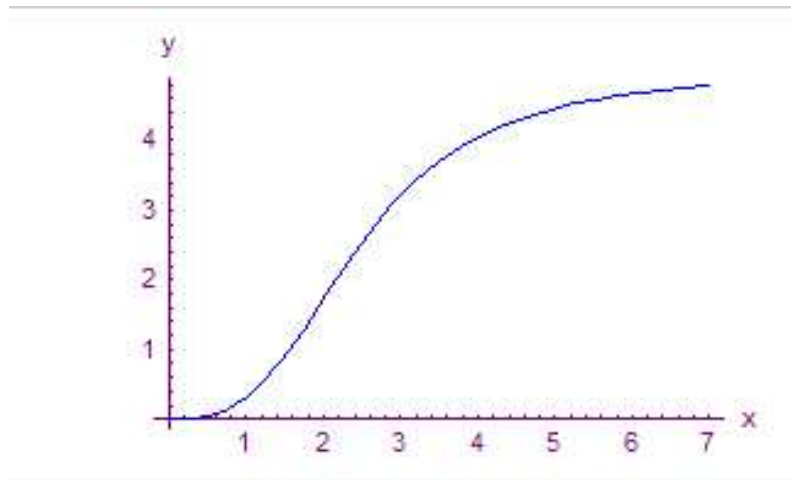
Die Konzentration der Lösung ist proportional zum Partialdruck. Daher existiert ein dynamisches Gleichgewicht.

Wenn ein Gas **keine einfache Lösung** (z.B.: Sauerstoff im Blut) bildet gilt:

$$c = C(P)$$

Hier handelt es sich um keine lineare, sondern um eine streng monoton wachsende Funktion.

Auf Grund dessen gibt es eine Umkehrfunktion ($P = C^{-1}(c)$).



Gastransport in einem Alveolus

$$V_A(c_I - c_A) = Q(c_a - c_v)$$

Dieses Gesetz verdeutlicht das Gleichgewicht zwischen der Anzahl der Gasmoleküle, die pro Zeiteinheit, aus der Atemluft entnommen bzw. hinzugefügt werden und der Anzahl jener Gasmoleküle, die pro Zeiteinheit, vom Blut aufgenommen bzw. abgegeben werden.

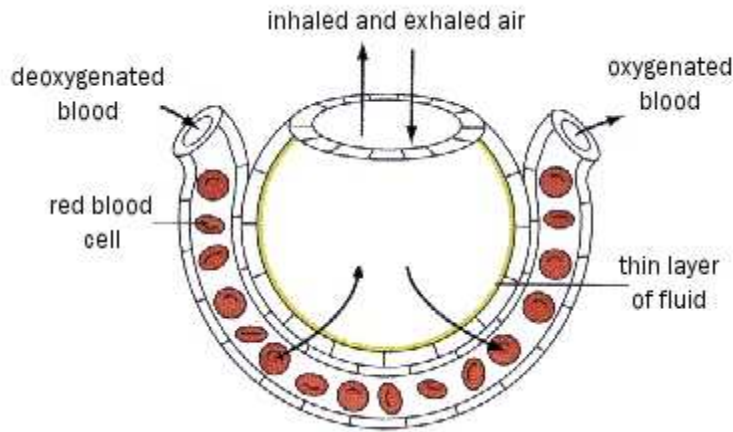
Ventilations-Perfusions-Verhältnis:

$$r = \frac{V_A}{Q}$$

Das Ventilations-Perfusions-Verhältnis beschreibt die Gaskonzentration in der alveolaren Luft und im arteriellen Blut.

Wenn die O_2 -Konzentration in der Außenluft absinkt, verlangsamt der Körper reflexartig den Blutfluss (Q). So kommt es auch zu einem Rückstau des Blutes aus dem Herzen und einem Druckanstieg in der Perfusion. Wenn dieser zu groß ist, kann es zum Eindringen von Blut in die Lungenbläschen kommen (=lebensgefährlich).

Wenn umgekehrt der Partialdruck von O_2 steigt, wird die Perfusion angekurbelt. Dies kann allerdings für den Menschen auch gefährlich sein.



Gasfluss:

= Anzahl der Moleküle, die pro Zeiteinheit aus der alveolaren Luft ins Blut kommen bzw. aus dem Blut in die alveolare Luft wechseln:

$$f = \frac{Q_r (\sigma k T c_l - c_v)}{r + \sigma k T} = Q \sigma (P_l - P_v) \frac{r}{r + \sigma k T}$$

Der Gasfluss ist wesentlich von r abhängig.

Je größer die Temperatur desto kleiner ist f (z.B.: bei Fieber)

Während die maximalen Volumen der Alveoli in der ganzen Lunge fast gleich sind, besitzen die Alveoli im unteren Bereich der Lunge, aufgrund des Gewichts des Lungengewebes, kleinere minimale Volumen.

Gastransport in der Lunge

Der Blutfluss durch die Lunge ist die Summe der Blutflüsse durch die Alveoli. Diese sind parallel geschaltet.

Parameter die für alle Alveoli gleich sind: k , T , σ , c_l , c_v ,

Parameter die von Alveoli zu Alveoli verschieden sind: Q_i , $(V_A)_i$, r_i , $(CA)_i$

Gesamtgasfluss in der Lunge:

$$f = \sigma (P_l - P_v) Q_0 E$$

E = Effizienz

$$E = \frac{1}{Q_0} \sum_{i=1}^{3 \times 10^8} \frac{Q_i r_i}{r_i + \sigma k T}$$

$$0 < E < 1$$

O2 Transport in der Lunge

Sauerstoff im Blut ist keine einfache Lösung und somit auch nicht mit einer linearen Funktion darstellbar.

Partialdruck von Sauerstoff im Blut:

$$P = C^{-1}(c) = H(c)$$

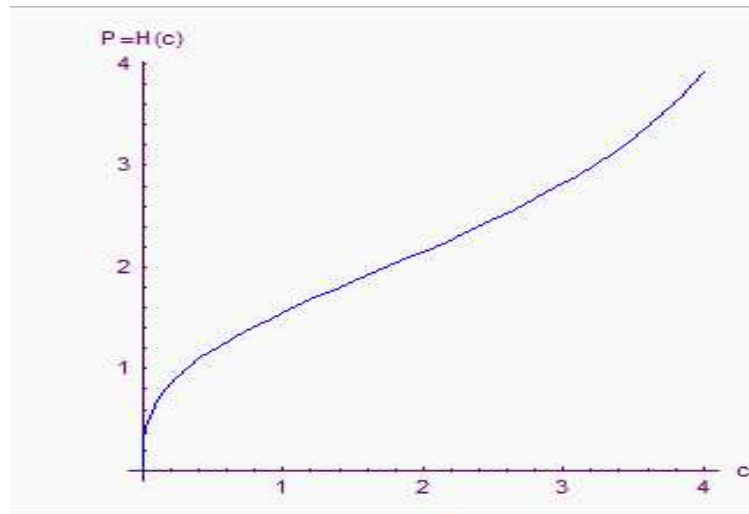
Die Funktion $H(c)$ ist nur schwer explizit darstellbar und muss abgeschätzt werden.

Eigenschaften von H :

$$H(0) = 0$$

$$H'(c) > 0 \text{ für } c \geq 0$$

$$H''(c) > 0 \text{ für } c \geq c_0$$



O2 Transport f in der Lunge:

$$f = \sum_i f_i = \sum_i Q_i (c_a(r_i) - c_v)$$

$$r_i = \frac{(V_A)_i}{Q_i}$$

Die c_a Funktion ist nicht explizit darstellbar.

Problem:

Stelle die c_a Funktion dar!

Überlegungen:

Für jedes r gibt es ein ganz bestimmtes c_a :
$$rc_I + c_v = c_a + \frac{r}{kT} H(c_a)$$

Diese Funktion ist beschränkt und streng monoton steigend.

Problembehandlung

1. Problemstellung:

Zeige, dass der Gasfluss (f) bei gegebenem Druck P_I und P_V maximal ist, wenn das Ventilations- Perfusions- Verhältnis für alle Alveoli r_i ($i=1, \dots, 3 \cdot 10^8$) gleich r_o ist, wobei r_o das Ventilations- Perfusions- Verhältnis der Gesamtlunge ist.

Ausgehend von der Gleichung für die Effizienz der Lunge bzw. des Gesamtgasaustausches,

$$f = \sum_{i=1}^N \dot{V}_i = Q_o \sigma \cdot (P_I - P_V) \cdot E$$

wobei der Faktor E für die Effizienz des Gasflusses steht,

$$E = \frac{1}{Q_o} \cdot \sum_{i=1}^N \frac{Q_i r_i}{r_i + \sigma k T}$$

wird zuerst die Funktion E näher betrachtet. Um ihre Eigenschaften näher zu bestimmen ohne die Funktion selber genau zu kennen.

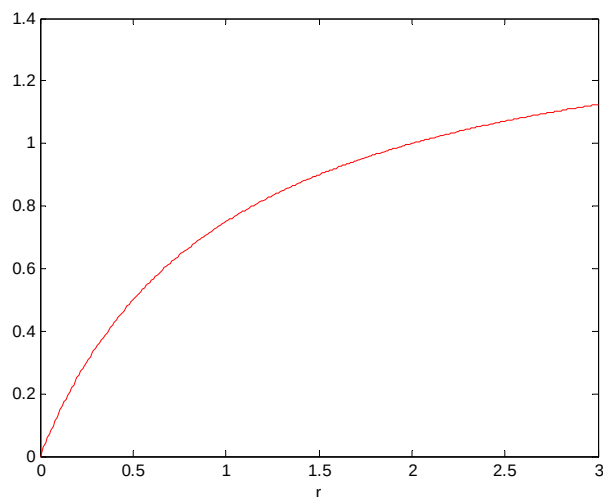
Vorhergehensweise:

- a. Durch Diskussion der Funktion wird zunächst ihre Gestalt näher bestimmt.

$$g_i(0)=0$$

$$g_i'(x)>0$$

$$g_i''(x)<0$$



Ergebnis:

Es handelt sich um eine streng monoton steigende Funktion.

- b. Im nächsten Schritt macht man sich zunutze, dass die Graphen der Funktionen (g_i) unterhalb jeder Tangente an die Kurve g_i liegen. Daher können wir die Funktionswerte $g_i(r_i)$ durch die Werte der Tangente an die Kurve von g_i im Punkt r_o nach oben abschätzen, d.h. es gilt $g_i(r_i) < y_i = g_i(r_o) + g_i'(r_o)(r_i - r_o)$, falls $r_i \neq r_o$ ist. Wenn man daher in der Formel für E die Funktionswerte $g_i(r_i)$ durch y_i ersetzt, erhält man eine Abschätzung von E nach oben:

$$E < \frac{1}{Q_o} \sum_{i=1}^N \frac{Q_i r_o}{r_o + \sigma k T} + \frac{1}{Q_o} \sum_{i=1}^N g_i'(r_o)(r_i - r_o) = \frac{r_o}{r_o + \sigma k T} = E_{\max}.$$

Man beachte, dass der zweite Summand auf der rechten Seite Null ist. Weiters sieht man, dass E genau dann den Wert E_{max} annimmt, wenn für alle Alveoli $r_i = r_o$ gilt.

Ergebnis:

Ein maximaler Gasfluss (f) kann nur erreicht werden, wenn das Ventilations-Perfusions-Verhältnis in allen Alveoli gleich ist:

$$f = Q_o \sigma \cdot (P_I - P_v) \cdot \frac{r_o}{r_o + \sigma k T} = f_{\max}$$

2. Problemstellung:

Zeige das analoge Resultat für den O₂ Transport in der Lunge.

Es ist zu beachten, dass O₂ im Blut keine einfache Lösung bildet d.h. es finden durch die Bindung ans Hämoglobin chemische Reaktionen statt. Ausgehend von dieser Formel für den Gasaustausch der Gesamtlunge

$$f = \sum_{i=1}^N Q_i \cdot [c_a(r_i) - c_v]$$

muss man im Fall von O₂ jedoch von einer nichtlinearen Beziehung ausgehen, die durch die Funktion H(c) beschrieben wird, wobei nur ein gewisser Bereich für den O₂ Partialdruck in der Lunge relevant ist ($c \geq c_o > 0$). Der Partialdruck von O₂ im Blut kann mit einer Umkehrfunktion H(c_a) beschrieben werden, die folgende Eigenschaften hat:

$$H(0) = 0$$

$$H'(c) > 0 \quad \text{wobei gilt } c \geq 0$$

$$H''(c) < 0 \quad \text{wobei gilt } c \geq c_o$$

Vorgehensweise:

- Durch einige Umformungen gingen wir zuerst von folgender Gleichung aus

$$r c_I + c_v = c_a + \frac{r}{k T} \cdot H(c_a)$$

um im nächsten Schritt die Funktion H(c_a) bzw. deren Abhängigkeit von der arteriellen O₂ Konzentration genauer zu betrachten. Weiters sieht man, dass es für ein fixes $r \geq 0$ genau eine Lösung $c_a = c_a(r) \leq L$ gibt. Man kann jetzt davon ausgehen, dass c_a für alle Alveoli gleich ist, da es sich bei den restlichen

Komponenten in der Gleichung um Konstanten handelt (c_l, c_v, k, T) die ebenfalls für alle Alveoli gleich sind.

Durch anschließendes Differenzieren konnten wir ohne explizite Darstellung der Funktion auf bestimmte Eigenschaften schließen. Im Fall von Sauerstoff können wir die Funktionswerte $c_a(r_i)$ durch die Werte der Tangente an die Kurve von y_i im Punkt r_o abermals nach oben abschätzen. d.h. es gilt $c_a(r_i) < y_i = c_a(r_o) + c_a'(r_o)(r_i - r_o)$, falls $r_i \neq r_o$ ist.

Wenn man wieder davon ausgeht, dass $r_i = r_o$, dann kommt man durch das Aufspalten der Summen und anschließendes Substituieren zu folgendem **Ergebnis:**

$$f = \sum_{i=1}^N Q_i (c_a(r_i) - c_v) \leq Q_o (c_a(r_o) - c_v) = E_{\max}$$

was bedeutet, dass wenn alle $r_i = r_o$, dann ist f gleich eine obere Schranke und damit E-max.

3. Problemstellung:

Aufgabe:

Die Lunge besteht aus 2 Teilen, in denen r_i (= Ventilations-Perfusions-Verhältnis) für alle Alveoli gleich ist. ($r_i = r_o$)

Es werden folgende Annahmen getroffen: $PI = 0$ (beschreibt Abatmen von Anestesiegasen)

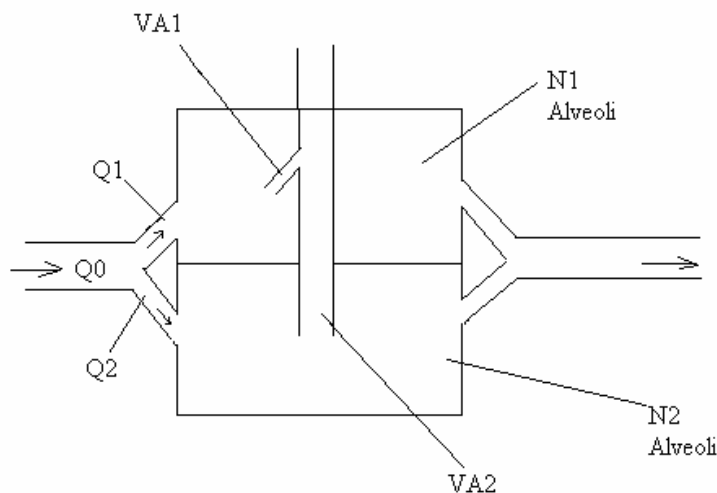
$$\delta kT = 1$$

$$Q_1 = 2 \text{ l/min}$$

$$Q_2 = 3 \text{ l/min}$$

$$Q_0 = Q_1 + Q_2$$

$$(VA)_0 = (VA)_1 + (VA)_2 = 5 \text{ l/min}$$



- a. Man sucht ein $(V_A)_1$ bei dem E (=Effizienz der Lunge) maximal wird (maximaler Gasaustausch). E soll als Funktion von $(V_A)_1$ dargestellt werden. Der Einfachheit halber nehmen wir für $(V_A)_1 = x$ an

Berechnung mit einer einfachen Lösung:

I. Allgemeine Lösung:

$$f = \sigma Q_0 (P_I - P_v) E$$

$$E = \frac{1}{Q_0} \left(\sum_{i=1}^{N_1} \frac{Q_i r_i}{r_i + 1} + \sum_{i=N_1+1}^{N_1+N_2} \frac{Q_i r_i}{r_i + 1} \right)$$

$$\tilde{r}_1 = \frac{(V_A)_1}{Q_1} = \frac{x}{Q_1} \quad , \quad \tilde{r} = \frac{(V_A)_2}{Q_2} = \frac{(V_A)_0 - x}{Q_2}$$

$$E = \frac{1}{Q_0} \left(\frac{x}{Q_1 + x} \cdot Q_1 + \frac{(V_A)_0 - x}{Q_2 + (V_A)_0 - x} \cdot Q_2 \right)$$

Es handelt sich um ein Extremwertbeispiel. Wir bilden die 1. Ableitung und setzen diese 0. So erhält man eine quadratische Gleichung und deshalb 2 mögliche Lösungen.

$$(Q_2 + (V_A)_0 - x) \cdot Q_1 = \pm (Q_1 + x) \cdot Q_2$$

1. Fall ergibt:

$$x = Q_1 \frac{(V_A)_0}{Q_0}$$

Im 2. Fall gibt es keine Lösung.

Wie wir bereits bei Problem 1 festgestellt haben, erreicht die Lunge die maximale Effizienz dann, wenn alle r_i (Ventilations-Perfusions-Verhältnis) für alle Alveoli gleich (r_0) ist.

- b. Es existiert ein $(V_A)_1$ bei dem die maximale Effizienz erreicht werden kann. Überraschender Weise kann man bei einer Lunge, bei der ein Teil schlechter arbeitet, auch eine maximale Effizienz erreichen.

I. Lösung mit oben angegebenen Zahlenwerten:

$$E(x) = \frac{2}{5} \cdot \frac{x}{x+2} + \frac{3}{5} \cdot \frac{5-x}{8-x}$$

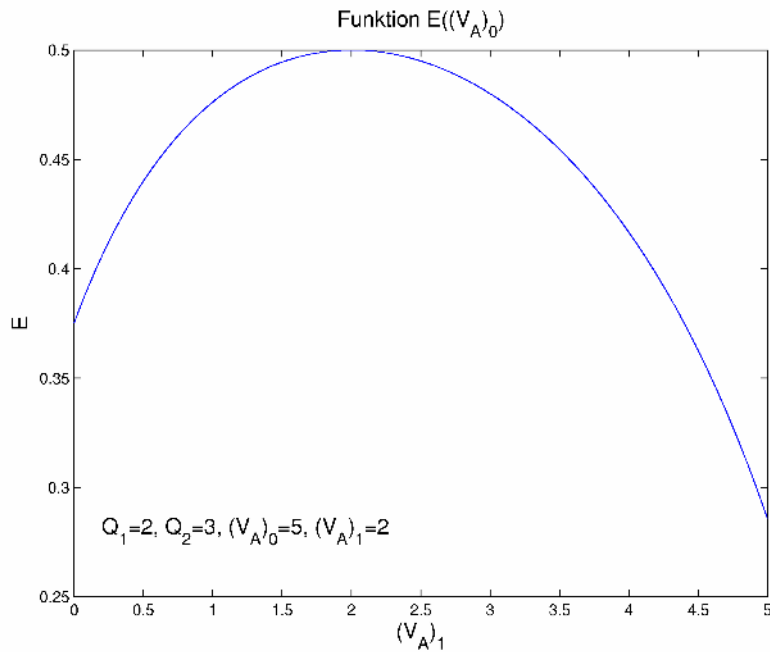
$$E'(x) = \frac{2}{5} \cdot \frac{2}{(x+2)^2} + \frac{3}{5} \cdot \frac{-3}{(8-x)^2} \quad , \quad E'(x) = 0$$

$$x^2 + 20x - 44 = 0$$

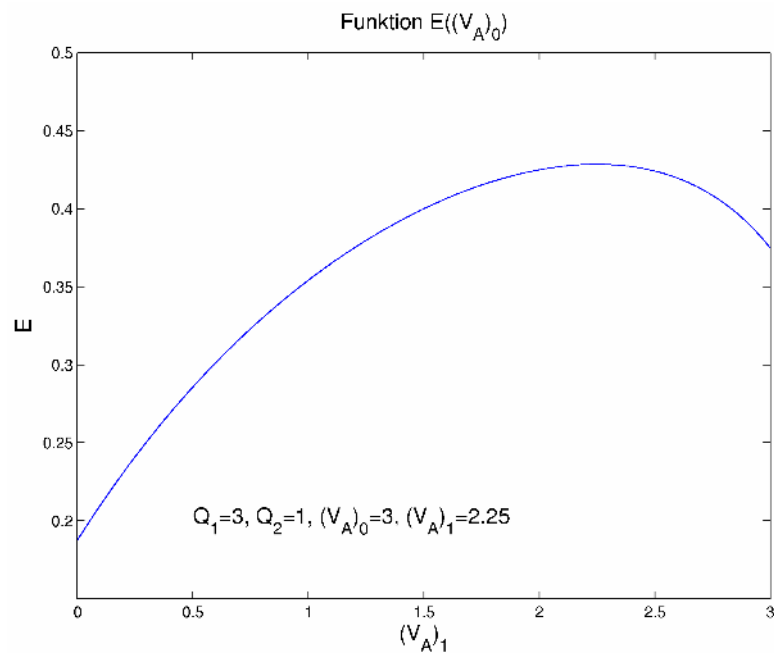
$$x_1 = 2$$

$(x_2 = -22)$ ($\rightarrow$ unmöglich da negativ)

$$E(2) = \frac{1}{2} \quad \rightarrow \text{Maximale Effizienz} = \frac{1}{2}$$



Diese Kurve stellt die $E(V_A)_1$ Funktion mit den gerechneten Zahlenwerten dar.



Hierbei handelt es sich ebenso um eine $E(V_A)_1$ Funktion mit anderen Zahlenwerten. Man kann sehr gut erkennen, dass es auch hier ein Maximum gibt.

c.

Ersetzt man in folgender Gleichung $\bar{P}_a = \frac{\sum_{i=1}^2 Q_i \cdot (c_a)_i}{Q_0 \cdot \sigma}$ den Blutfluss Q_i durch $\frac{V_i}{t}$ und

die Konzentration im arteriellen Blut $(c_a)_i$ durch $\frac{n_i}{V_i}$, so kürzt sich die Gleichung

folgendermaßen: $\bar{P}_a = \frac{\sum_{i=1}^2 \frac{n_i}{t}}{Q_0 \cdot \sigma}$. Q_0 wird durch $\frac{V_0}{t}$ ersetzt und man erhält folgenden

Doppelbruch: $\bar{P}_a = \frac{\sum_{i=1}^2 \frac{n_i}{t}}{\frac{V_0}{t} \cdot \sigma}$. Durch Kürzen ergibt sich: $\bar{P}_a = \frac{\sum_{i=1}^2 n_i}{V_0 \cdot \sigma}$, wobei man durch

Ersetzen von $\frac{\sum_{i=1}^2 n_i}{V_0}$ durch $\sum_{i=1}^2 c_i$ folgende Gleichung erhält: $\bar{P}_a = \frac{\sum_{i=1}^n c_i}{\sigma}$ (Man beachte

die Relation zu $P_j = \frac{c_j}{\sigma_j}$).

Interpretiert man die nun erhaltene Gleichung, lässt sich feststellen, dass der durchschnittliche Partialdruck der Quotient aus der Summe der i Teile und der Löslichkeit ist.

Ersetzt man in der Gleichung $\bar{P}_A = kT \frac{\sum_{i=1}^2 (V_A)_i \cdot (c_A)_i}{(V_A)_0}$ die Ventilationsflüsse durch

$\frac{(Vol_A)_i}{t}$ bzw. $\frac{(Vol_A)_0}{t}$ sowie die Konzentration durch $\frac{(n_A)_i}{(Vol_A)_i}$ so erhält man:

$\bar{P}_A = \frac{\sum_{i=1}^2 \frac{(Vol_A)_i}{t} \cdot \frac{(n_A)_i}{(Vol_A)_i}}{\frac{(Vol_A)_0}{t}}$. Gekürzt ergibt sich: $\bar{P}_A = kT \frac{\sum_{i=1}^2 (n_A)_i}{(Vol_A)_0}$.

Projekt: Biomechanik

Weitsprung mit Zusatzgewichten

Betreuerin: Dr. Sigrid Thaller

TeilnehmerInnen: Claudia Andrä, Romana Boiger, Thomas Janisch, Theodora Korkas, Philipp Prinz und Elias Windisch

Problemstellung

Der Weitsprung aus dem Stand war Teil des antiken olympischen Fünfkampfes. Die Sportler verwendeten beim Sprung Zusatzgewichte, so genannte Halteres, die sie in den Händen hielten. Diese Gewichte sollten die Sprungweite erhöhen. Wie lässt sich die Wirkung der Sprunggewichte erklären? Welche Größe der Gewichte wirkt sich günstig auf die Bewegung aus? Sind die Effekte für alle Sportler gleich? Mit Hilfe von schon bestehenden Modellen einfacherer Bewegungen soll die Wirkungsweise der Zusatzgewichte beim Standweitsprung erklärt und die Bewegung am Computer simuliert werden.

Geschichte

Im antiken Griechenland war der Pentathlon ein bedeutender Wettkampf. Er bestand aus Laufen, Diskuswerfen, Speerwerfen, Ringen und Standweitsprung. Bei diesen Wettbewerben wurden herausragende Leistungen erzielt und etliche Weltrekorde aufgestellt. Besonders stach hier der Standweitsprung hervor: Im 5. Jh. v. Chr. gelang es dem griechischen Athleten Phayllos, im Standweitsprung eine Weite von 16,28 m zu erzielen.



Abb.1 © Museum of Fine Arts Boston

Während heute fast alle antiken Rekorde übertroffen worden sind, bleibt dieser Weitenrekord noch immer unerreicht und aus verschiedenen Gründen ein Rätsel. Ein Grund besteht in der Tatsache, dass die Sprungweite keines Sportlers der Neuzeit 9m im Weitsprung übertrifft. Aus diesem Grund waren Wissenschaftler umso mehr über jene gewaltige Weite erstaunt. Man nimmt mittlerweile jedoch an, dass die alten Griechen nicht einen, sondern vermutlich fünf Sprünge hintereinander tätigten, was allerdings noch nicht erwiesen ist.

Aufgrund von Sprungdarstellungen auf alten Vasen weiß man mittlerweile, dass die Athleten Gewichte (so genannte „Halteres“) zur Verbesserung ihrer Sprungweite zu Hilfe nahmen. Aber die Art, wie die Sprünge tatsächlich vollzogen wurden und wie die Gewichte den Athleten beim Weitsprung wirklich unterstützten, stand lange im Mittelpunkt vieler Diskussionen [1].

In einem Versuch von M. Lenoir, D. De Clercq und W. Laporte [2] sprangen vier Athleten mit und ohne Gewichte fünfmal hintereinander. Die Ergebnisse zeigten, dass Gewichte die Sprungweite tatsächlich beeinflussen:

Ohne Gewichte	Mit Gewichten
$13.88 \pm 0.7 \text{ m}$	$14,64 \pm 0.76 \text{ m}$

Modell für den senkrechten Sprung ohne Ausholbewegung

Die Modellgleichungen für den senkrechten Sprung haben wir von [3] übernommen.

1) Einschaltfunktion $s(t)$

Jeder Muskel besteht aus mehreren Muskelfasern und ein Nerv versorgt eine Gruppe von Muskelfasern. Diese bekommt den Befehl zu kontrahieren oder nicht (eine motorische Einheit kann nicht „halb“ kontrahieren). Damit eine Bewegung zustande kommt, müssen motorische Einheiten eingeschaltet werden.

$n(t)$ Anzahl der schon in Betrieb genommenen motorischen Einheiten

$n_{\max}$ max. Anzahl der gleichzeitig einschaltbaren motorischen Einheiten

$$s(t) = 1 - e^{-At}$$

z.B.: A ist 7s^{-1} .

Herleitung der Formel für die Einschaltfunktion $s(t) = 1 - e^{-At}$:

$$\begin{aligned} \frac{dn(t)}{dt} &\sim n_{\max} - n(t) \\ \frac{dn(t)}{dt} &= A \cdot (n_{\max} - n(t)) \\ n(t) &= s(t) \cdot n_{\max} \\ n(t) &= n_{\max} (1 - e^{-A(t-t_0)}) \\ n(t)' &= -n_{\max} \cdot e^{-A(t-t_0)} \cdot (-A) \\ (-n_{\max} \cdot e^{-A(t-t_0)}) \cdot (-A) &= A \cdot (n_{\max} - n_{\max} \cdot (1 - e^{-A(t-t_0)})) \\ n_{\max} \cdot e^{-A(t-t_0)} &= n_{\max} \cdot e^{-A(t-t_0)} \end{aligned}$$

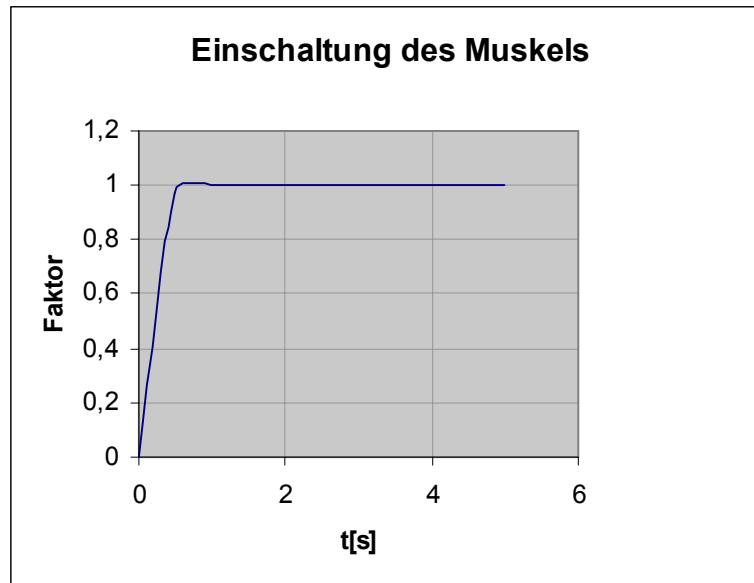


Abb.2: Einschaltfunktion

2) Muskelkraft $f(v)$

Die Muskelkraft verhält sich indirekt proportional zur Kontraktionsgeschwindigkeit. Je größer diese Geschwindigkeit ist, desto geringer ist die Muskelkraft. Die rechteckige Fläche zwischen der Kurve und der x-Achse gibt die Leistung des Muskels an (siehe Abbildung). Anfangs vergrößert sich die Leistung mit zunehmender Geschwindigkeit, bis sie ein Maximum erreicht. Eine höhere Geschwindigkeit wirkt sich negativ auf die Leistung aus.

$$f = \frac{c}{v + b} - a$$

f Kraft

v Kontraktionsgeschwindigkeit

a, b, c Konstante, Eigenschaften für einen speziellen Muskel

z.B.:

$a = 3069 \text{ N}$

$b = 0,1074 \text{ m/s}$

$c = 2478 \text{ W}$

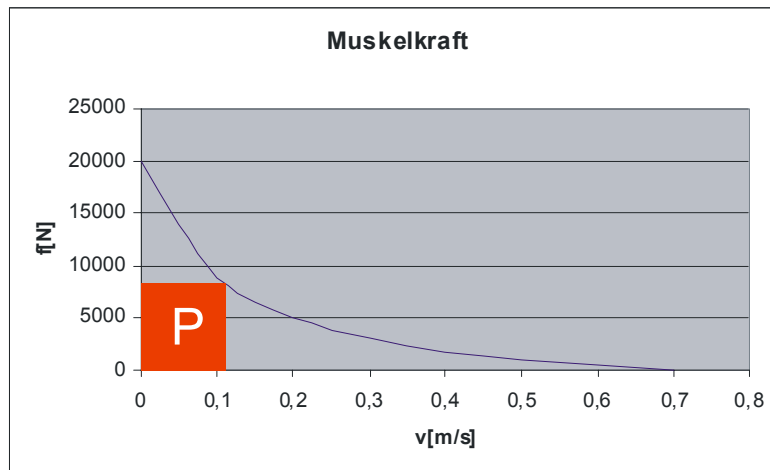


Abb. 3: Kraft im Muskel als Funktion der Geschwindigkeit

Kraft-Geschwindigkeitsrelation

$y = 1/x$ skalieren und verschieben $\rightarrow$ $y = \frac{c}{x+b} - a$ $x \rightarrow v$ $y \rightarrow f$ $f = \frac{c}{v+b} - a$	
Diese Kraft-Geschwindigkeitsrelation gilt nur für konzentrische Bewegungen (v größer oder gleich 0).	
$f_{iso} = \frac{c}{b} - a$	f_{iso} ... maximale Kraft (bei $v=0$)
$v_{max} = \frac{c}{a} - b$	v_{max} ... maximale Geschwindigkeit (bei $f=0$)
$p = \frac{w}{t} = f \cdot v$	

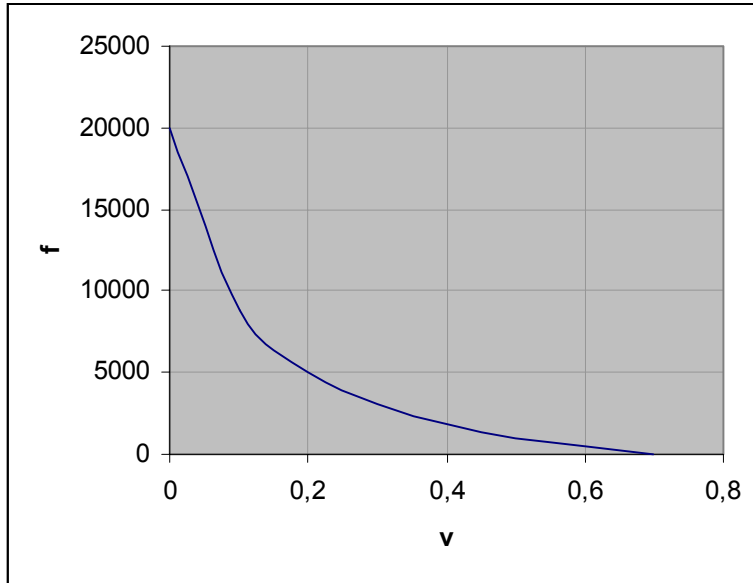


Abb. 5

$$p = f \cdot v$$

$$p = \frac{cv}{v+b} - av$$

$$p' = \frac{bc}{(v+b)^2} - a$$

$$\frac{bc}{(v+b)^2} - a = 0$$

$$v_{opt} = \sqrt{\frac{bc}{a}} - b$$

$$f_{opt} = \sqrt{\frac{ac}{b}} - a$$

$$p_{max} = f_{opt} \cdot v_{opt}$$

$$p_{max} = ab + c - 2\sqrt{abc}$$

3) Geometriefunktion G(X)

Es wird das Verhältnis zwischen der Kraft im Muskel und der Kraft, die außen wirkt, beschrieben. Dabei spielen bestimmte Faktoren, die bei jedem Menschen unterschiedlich sind, eine Rolle: Länge des Ober- und Unterschenkels, der Knieradius, die Muskelansätze und die verschiedenen Winkel.

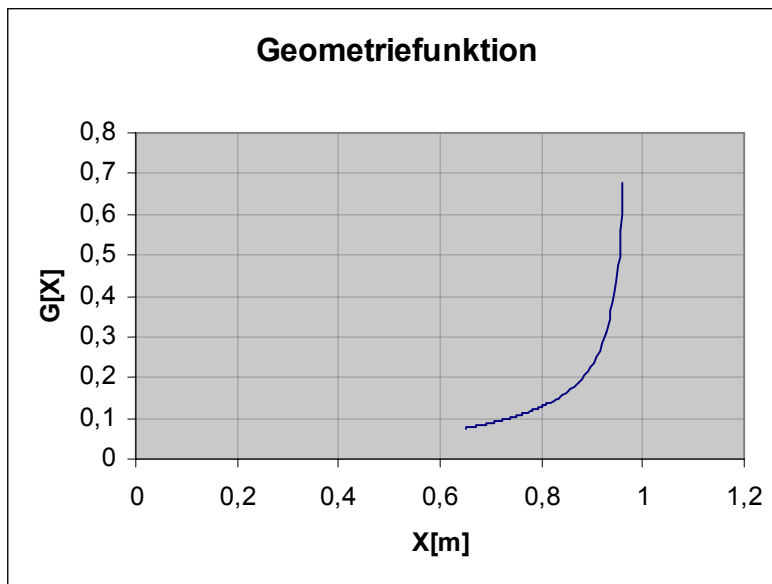


Abb.6: Geometriefunktion mit Anfangsort = 0,65 m

$$\begin{aligned} \text{I.} \\ X^2 &= lo^2 + lu^2 - 2 \cdot lo \cdot lu \cdot \cos \sigma \\ \text{II.} \\ \sigma &= 2\beta + \arcsin\left(\frac{r}{ko} \sin \beta\right) + \arcsin\left(\frac{r}{ku} \sin \beta\right) \\ \text{III.} \\ G(X) &= \frac{r \cdot \sin \beta}{lo \cdot lu \cdot \sin \sigma} X \end{aligned}$$

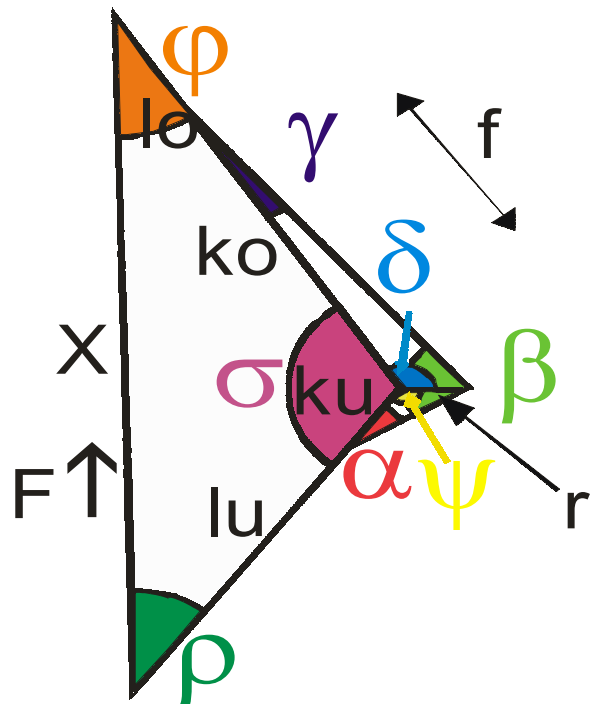


Abb.7 Skizze eines Beines: lo...Oberschenkel, lu...Unterschenkel, σ...Kniewinkel, ko,ku... Ursprung und Ansatz des Muskels, r...Knieradius.

Beweis der Gleichungen:

- Ad I.

Kosinussatz

- Ad II.

$$\begin{aligned} \frac{ko}{\sin \beta} &= \frac{r}{\sin \gamma} \\ \gamma &= \arcsin\left(\frac{r \cdot \sin \beta}{ko}\right) \end{aligned}$$

$$\begin{aligned} \frac{ku}{\sin \beta} &= \frac{r}{\sin \alpha} \\ \alpha &= \arcsin\left(\frac{r \cdot \sin \beta}{ku}\right) \end{aligned}$$

$$\sigma = 2\beta + \gamma + \alpha$$

$$\delta = 180 - \beta - \gamma$$

$$\psi = 180 - \beta - \alpha$$

$$360 - 2\beta - \gamma - \alpha + \sigma = 360$$

$$\sigma = 2\beta + \gamma + \alpha$$

- Ad III.

Mit Hilfe des Drehmoments

$$\vec{D} = |\vec{r} \times \vec{F}| = |\vec{r}| \cdot |\vec{F}| \cdot \sin \alpha$$

Bei Geschwindigkeit 0 muss die Summe aller im Knie wirkenden Drehmomente gleich 0 sein.

$\vec{D} = \vec{lo} \cdot \vec{F} \cdot \sin \varphi$	$G(X) \frac{lo \cdot \sin \varphi}{r \cdot \sin \beta} = 0$
$\vec{D} = \vec{r} \cdot \vec{f} \cdot \sin \beta$	$G(X) = \frac{r \cdot \sin \beta}{lo \cdot \sin \varphi}$
$ \vec{lo} \cdot \vec{F} \cdot \sin \varphi = \vec{r} \cdot \vec{f} \cdot \sin \beta$	$\frac{lu}{\sin \varphi} = \frac{x}{\sin \sigma}$
$G(X) = \frac{ \vec{F} }{ \vec{f} }$	$\sin \varphi = \frac{lu \cdot \sin \sigma}{x}$

$$G(X) = \frac{r \cdot \sin \beta}{lo \cdot lu \cdot \sin \sigma} X$$

4) Bewegungsgleichung

Laut Newton bewirkt die Summe aller Kräfte die Beschleunigung des Körpers. Die Kräfte setzen sich zusammen aus dem Gewicht des Körpers und die vom Muskel verursachte Kraft. Dabei müssen die Einschaltfunktion und ein Faktor für die Geometrie berücksichtigt werden.

$$m \cdot a = \Sigma \cdot F_i = -m \cdot g + F_m = -m \cdot g + s(t) \cdot f(v) \cdot G(X)$$

X = Ort des Schwerpunkts

F_m = vom Muskel erzeugte Kräfte

g = Gravitationsbeschleunigung

v = Kontraktionsgeschwindigkeit

$G(X)$ = Geometriefunktion

$s(t)$ = Einschaltfunktion

t = Zeit

a = Beschleunigung

f = Muskelkraft

m = Masse

Wie wirkt eine Armbewegung?

- Ohne Gewichte
- Mit Gewichten

Versuchsaufbau: Waage (analog), Gewichte (zweimal 1 kg)

Aufgabe: Die Arme so schnell wie möglich in die Luft reißen und die Gewichtsänderung auf der Waage feststellen

Name	Eigengewicht (N)	Zusatzgewicht (N)	Kraft erzeugt durch Armbewegung (N)	Gewichtsänderung (N)
Philipp	667,08	0	784,8	117,72
Philipp	667,08	19,62	882,9	235,44
Theo	470,88	0	588,6	117,72
Theo	470,88	19,62	657,27	206,01
Elias	735,75	0	981	245,25
Elias	735,75	19,62	981	264,87
Romana	568,98	0	686,7	117,72
Romana	568,98	19,62	686,7	137,34
Thomas	676,89	0	833,85	156,96
Thomas	676,89	19,62	931,95	274,68
Claudia	549,36	0	686,7	137,34
Claudia	549,36	19,62	716,13	166,77

Claudia wurde im Durchschnitt nicht berücksichtigt

a) Ohne Hanteln:

Im Durchschnitt gab es eine Gewichtsänderung von 151,074 N bei einer Armbewegung nach oben bei maximaler Geschwindigkeit.

b) Mit Hanteln:

Durchschnittlich gab es eine Gewichtsänderung von 223,668 N bei einer Armbewegung nach oben mit jeweils einem Gewicht von 1kg bei maximaler Geschwindigkeit.

Simulationsprogramm

Die folgenden Simulationen des senkrechten Sprunges wurden mit [4] durchgeführt.

The screenshot shows the input window of a simulation program. It is divided into two main sections: 'Muskeleigenschaften' (Muscle Properties) and 'Muskeleigenschaften (Hillsche Parameter)' (Muscle Properties (Hill's Parameters)).

Muskeleigenschaften:

- Isometrische Kraft [N]: 20000
- Maximal mögliche Kontraktionsgeschwindigkeit [m/s]: 0.7
- Maximal mögliche Leistung [W]: 1000
- Masse [kg]: 30
- Parameter der Schaltfunktion [1/s]: 500
- Gravitationsbeschleunigung [m/s²]: 9.81
- Anfangsort [m]: 0.87364
- Anfangsgeschwindigkeit [m/s]: 2.56121

Muskeleigenschaften (Hillsche Parameter):

- Hillscher Parameter a [N]: 3069.04
- Hillscher Parameter b [m/s]: 0.1074
- Hillscher Parameter c [W]: 2478
- Länge des Oberschenkels [m]: 0.52
- Länge des Unterschenkels [m]: 0.45
- Abstand Muskelursprung - Kniegelenk [m]: 0.52
- Abstand Muskelansatz - Kniegelenk [m]: 0.06
- Kniegelenkradius [m]: 0.06

At the bottom right, there are two buttons: 'Defaults' and 'Start'.

Abb.8 Eingabefenster des Simulationsprogramms

Bei den folgenden Ergebnissen bezeichnet t_{end} die Zeit beim Absprung, V_{end} die Geschwindigkeit des Schwerpunkts beim Absprung, $\max(V)$ die größte Geschwindigkeit während des Sprunges, $\max(F)$ die größte auftretende Bodenreaktionskraft, $\max(P)$ die größte auftretende mechanische Leistung und h die größte Höhe des Schwerpunkts.

Standardwerte: Masse: 80 kg Gravitationsbeschleunigung $9,81 \text{ m/s}^2$ Anfangsort: 0,65 m	Ergebnisse: $t_{\text{end}} = 0.5 \text{ s}$ $V_{\text{end}} = 1.05 \text{ m/s}$ $\max(V) = 1.11 \text{ m/s}$ $\max(F) = 1027.79 \text{ N}$ $\max(P) = 974.44 \text{ W}$ $h = 1.01 \text{ m}$
--	--

Simulation1: Änderung: 40 kg Ergebnisse: $t_{\text{end}} = 0.32 \text{ s}$ $V_{\text{end}} = 1.59 \text{ m/s}$ $\max(V) = 1.64 \text{ m/s}$ $\max(F) = 687.63 \text{ N}$ $\max(P) = 862.74 \text{ W}$ $h = 1.07 \text{ m}$ Da die Masse geringer ist, benötigt man weniger Kraft zum Springen. Sowohl die Höhe als auch die Geschwindigkeit des Sprungs vergrößern sich.	Simulation2: Änderung: 110 kg Ergebnisse: $t_{\text{end}} = 0.7 \text{ s}$ $V_{\text{end}} = 0.77 \text{ m/s}$ $\max(V) = 0.86 \text{ m/s}$ $\max(F) = 1244.32 \text{ N}$ $\max(P) = 996.3 \text{ W}$ $h = 0.99 \text{ m}$ Aufgrund der größeren Masse benötigt man mehr Kraft zum Springen. Dadurch verringern sich die Sprunghöhe und die Absprungszeit.
Simulation3: Änderung: 0,45m Anfangsort Ergebnisse: $t_{\text{end}} = 1.12 \text{ s}$ $V_{\text{end}} = 1.1 \text{ m/s}$ $\max(V) = 1.16 \text{ m/s}$ $\max(F) = 939.54 \text{ N}$ $\max(P) = 999.73 \text{ W}$ $h = 1.01 \text{ m}$ Wegen des tieferen Schwerpunkts am Anfang kann man eine höhere Geschwindigkeit aufbringen und erreicht somit eine größere Sprunghöhe.	Simulation4: Änderung: 0,95 m Anfangsort Ergebnisse: $t_{\text{end}} = 0.08 \text{ s}$ $V_{\text{end}} = 0.39 \text{ m/s}$ $\max(V) = 0.39 \text{ m/s}$ $\max(F) = 1318.79 \text{ N}$ $\max(P) = 391.91 \text{ W}$ $h = 0.97 \text{ m}$ Durch den höheren Schwerpunkt am Anfang ist die Hocke nicht so tief und man kann somit nicht die gleiche Geschwindigkeit aufbringen, um dieselbe Höhe zu erreichen.

Simulation5: Änderung: Gravitation auf 1/6: $1,635 \text{ m/s}^2$ Ergebnisse: tend = 0.33 s Vend = 1.74 m/s max(V) = 1.74 m/s max(F) = 673.22 N max(P) = 852.89 W h = 1.85 m Da die Gravitation und damit das Gewicht verringert wird, ist es leichter, mit weniger Kraft in kürzerer Zeit höher zu springen.	Simulation6: Änderung: Gravitation auf das 1,5 fache: $14,715 \text{ m/s}^2$ Ergebnisse: tend = 0.74 s Vend = 0.66 m/s max(V) = 0.82 m/s max(F) = 1288.8 N max(P) = 997.85 W h = 0.98 m Da die Gravitation erhöht wird, ist es schwieriger hoch zu springen. Man benötigt daher mehr Zeit und man springt niedriger.
--	---

Zerlegung der Simulation in einzelne Zeitabschnitte

Wir wollen mit dem Simulationsprogramm während einer Bewegung verschiedene Gewichte berücksichtigen. Daher untersuchten wir, ob es möglich ist, die Rechnung in verschiedene Zeitabschnitte zu unterteilen. Das Programm rechnet allerdings für jeden neu begonnen Zeitabschnitt die Einschaltfunktion von neuem, obwohl der Muskel in der Realität schon mehr aktiviert ist, was zu Ungenauigkeiten führen kann. Daher untersuchten wir, unter welchen Bedingungen sich diese Fehler gering auswirken.

Durch einen geringeren Anfangsort (tiefere Hocke) und ein höheres Gewicht ist eine größere Kraft erforderlich und dadurch wird die Einschaltungszeit verkürzt. Der Faktor wandert gegen 1, was bedeutet, dass der Muskel fast eingeschaltet ist. Je mehr Zeit vergeht, desto näher wandert der Faktor gegen 1.

Beispiel: Masse = 110 kg
Anfangsort = 0.45 m
Anfangsgeschwindigkeit = 0 m/s

Nach 0.05 sek : Ort = 0,45005 m
 V = 0,00297 m/s
 F = 1090,87969 N

Absprung mit diesen Werten $\rightarrow F = 1079,1 \text{ N}$

Vergleich eines Sprunges ohne Armbewegung und mit Armbewegung

- Mit Hilfe des Simulationsprogramms [4]
- Mit Hilfe von Matlab

a) Simulationsprogramm [4]

Die aufgrund der Armbewegung vergrößert auftretende Kraft und Gewichtsänderung kann man nur dann berücksichtigen, wenn man die Gravitation dementsprechend verändert und die Masse konstant bleibt. Die Bewegungsgleichung lautet:

$$\begin{aligned} m \cdot a &= -m \cdot g + F_{Arm} + s(t) \cdot f(v) \cdot G(X) \\ &= -m \cdot g' + s(t) \cdot f(v) \cdot G(x) \end{aligned}$$

Wenn man die Formel für den senkrechten Sprung betrachtet, sieht man, dass sowohl auf der rechten als auch auf der linken Seite Masse vorkommt, daher ist es nicht zielführend, diese zu verändern, da sie sich sonst nicht mehr eliminieren lässt. Daher muss man einen anderen Weg finden, um die Gewichtsänderung zu berücksichtigen. Die zweite Größe, die neben der Masse in der Formel vorkommt, ist die der Gravitation. Wenn man die Gravitation ändert, ändert man somit auch das Gewicht. Das beste Beispiel dafür ist die Tatsache, dass man am Mond weniger Gewicht als auf der Erde hat, da die Gravitation niedriger ist. Die Masse bleibt dabei aber gleich.

Wenn die Gewichtsänderung nun beispielsweise 13 kg beträgt, so muss man diese zur Körpermasse addieren und durch die Körpermasse dividieren. Der dabei errechnete Wert wird mit der Erdgravitation von $9,81 \text{ m/s}^2$ multipliziert und man erhält die Gravitation mit Berücksichtigung der Kraftänderung:

Masse: 40 kg Gewichtsänderung: 13 kg Gravitation: $9,81 \text{ m/s}^2$	$\frac{53}{40} = 1,325 \longrightarrow 1,325 \times 9,81 = 12,99 \text{ m/s}^2$
--	---

Ohne Armbewegung: Ergebnisse: $t_{end} = 0.3 \text{ s}$ $V_{end} = 2.31 \text{ m/s}$ $\max(V) = 2.37 \text{ m/s}$ $\max(F) = 800.09 \text{ N}$ $\max(P) = 999.78 \text{ W}$ $\text{Integral } Fdt = 157.88 \text{ Ns}$ $\text{Integral } Pdt = 217.32 \text{ J}$ $h = 1.19 \text{ m}$	Mit Armbewegung: Ergebnis: $h = 1.49 \text{ m}$ <u>Rechenweg:</u> Teilung des Sprunges in 7 Teilstücke und, um die Gewichtsänderung durch die Armbewegung zu berücksichtigen, haben wir die Gravitation dementsprechend geändert (Masse bleibt gleich, nur Gravitation wird verändert). <u>Begründung:</u> Durch mehr Gewicht arbeitet man langsamer und kann dadurch mehr Kraft einsetzen und somit höher springen.
Mit Gewichten $h = 1,8 \text{ m}$ <u>Begründung:</u> Durch zusätzliche Gewichte arbeitet der Muskel langsamer und kann dadurch mehr Kraft einsetzen und somit höher springen. Es gibt allerdings eine Gewichtsgrenze, ab der die Sprunghöhe wieder abnimmt, weil man die Arme nicht mehr so schnell bzw. überhaupt nicht mehr nach oben reißen kann.	

b) Matlab

Durch das Heben der Arme wirkt eine Kraft nach unten, wodurch das Gewicht größer wird. Diese Kraft wird in diesem Modell durch eine Sinusfunktion dargestellt (siehe Abb.9)

$$y = -151.074 \sin(20\pi/3 \cdot x)$$

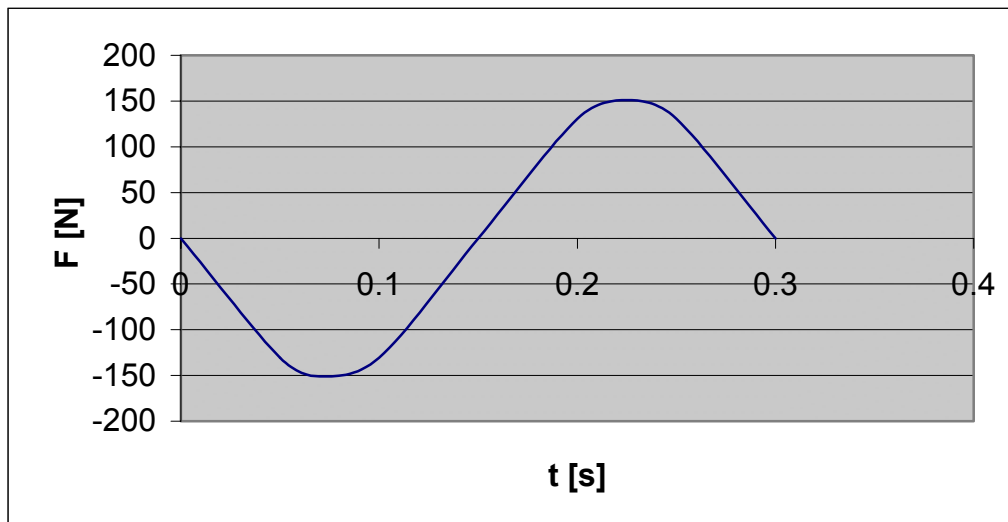


Abb. 9 : Kraft aufgrund der Armbewegung

Da die Armbewegung aber ca. ab der Hälfte wieder abgebremst wird, wirkt die Kraft in die entgegengesetzte Richtung, wodurch das Gewicht wieder abnimmt. Beim Absprung ist man dadurch leichter.

Die Abbildungen 10, 11 und 12 zeigen einen Vergleich der Schwerpunktsgeschwindigkeit V , Muskelkraft f_m und Bodenreaktionskraft F eines senkrechten Sprunges mit und ohne Armbewegungen.

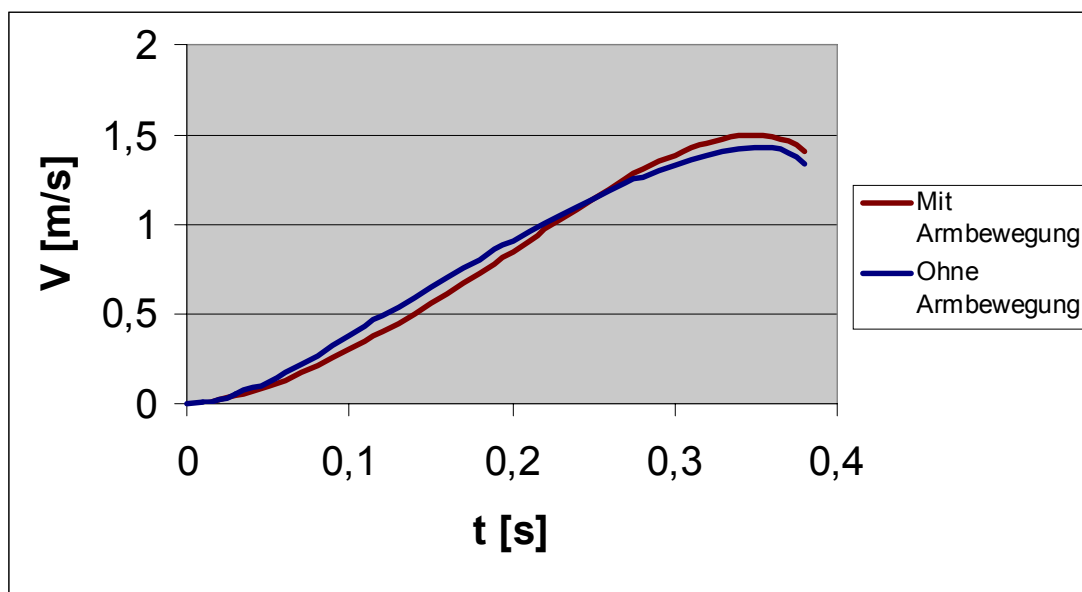


Abb. 10: Vergleich der Schwerpunktsgeschwindigkeit beim Sprung mit und ohne Armbewegung.

Zu Beginn eines Sprungs ohne Armbewegung ist die Geschwindigkeit gegenüber eines Sprungs mit Armbewegung höher, gegen Ende wird sie aber kleiner, weil man mit Armbewegung zuerst mehr Gewicht (weniger Beschleunigung) und später weniger Gewicht hat.

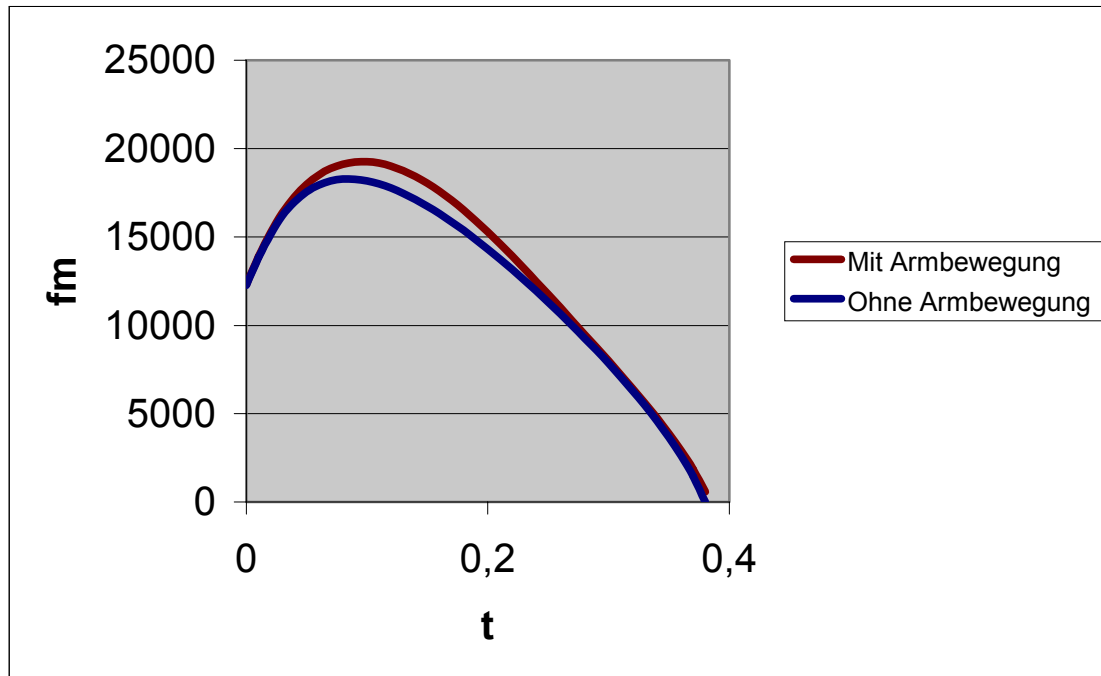


Abb. 11: Vergleich der Muskelkraft beim Sprung mit und ohne Armbewegung.

Da man bei einem Sprung mit Armbewegung am Anfang mehr Gewicht hat, kann man den Muskel nur langsamer bewegen und mehr Kraft einsetzen als bei einem Sprung ohne Armbewegung.

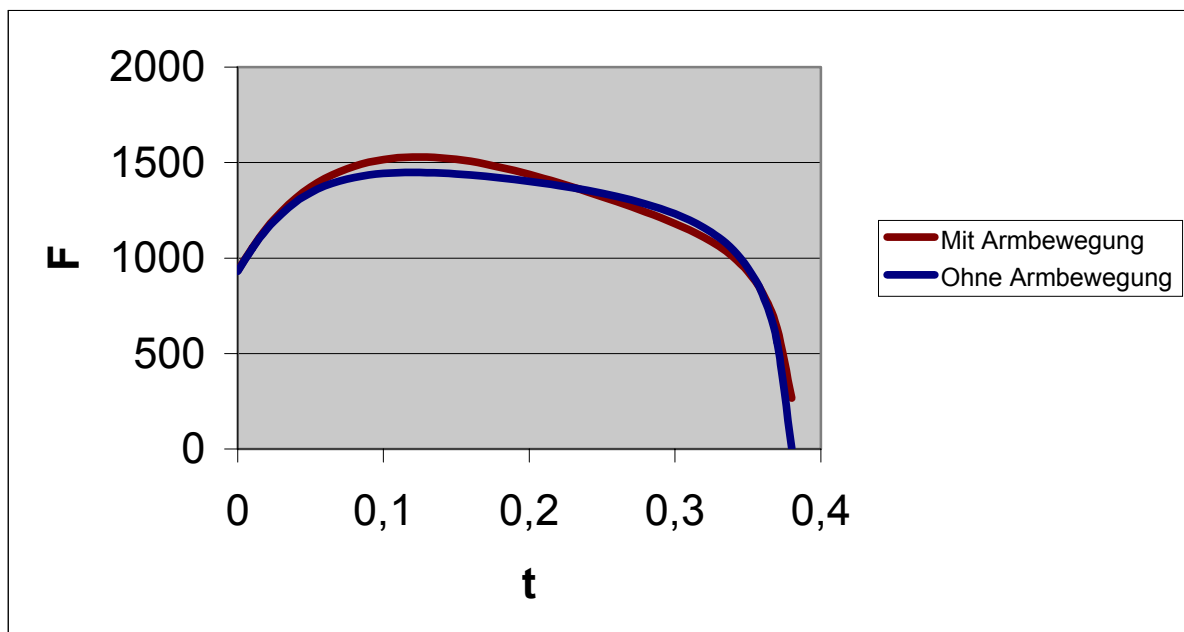


Abb. 12: Vergleich der Bodenreaktionskraft beim Sprung mit und ohne Armbewegung

Durch das Heben der Arme wirkt mehr Kraft auf den Boden, als ohne Armbewegung.

Versuch: Standweitsprung mit bzw. ohne Halteres

In einem praktischen Versuch wollten wir die Wirkung von Halteres feststellen. Zunächst sprang jeder von uns sowohl ohne Gewichte als auch ohne Armbewegung. Mit Armbewegung zeigte sich bei allen eine deutliche Erhöhung der Sprungweite, die teilweise sogar bis zu 33cm betrug.

Im Zuge des dritten Sprunges, bei dem beide Arme mit je 1 kg beschwert waren, stellte sich heraus, dass diese zusätzlichen Gewichte nicht für jeden eine Hilfe darstellten. So vergrößerte sich die Sprungweite bei einigen zwar (um bis zu 22 cm), bei anderen blieb sie jedoch etwa gleich oder verringerte sich sogar ein wenig.

Bei einem zusätzlichen Gewicht von 2 kg pro Arm wurde dieser Effekt noch deutlicher sichtbar. Einige sprangen weiter als zuvor, für andere war dieses Gewicht allerdings zu schwer, sodass sich ihre Sprungweite um bis zu 20 cm verkürzte.

Zwischen der Armbewegung und der Sprungweite lässt sich ein Zusammenhang feststellen: Durch das Zurückschwingen der Arme während des Sprunges wird der Schwerpunkt nach hinten verlagert und die Weite so erhöht. Sind die Gewichte allerdings zu schwer, gelingt dieses Zurückschwingen der Arme nicht mehr so gut oder gar nicht mehr und die Gewichte haben einen negativen Effekt. Dieser „Knackpunkt“ der zusätzlichen Gewichte ist nur individuell festlegbar, es gibt keinen „allgemeinen Grenzwert“.



Abb.13

	1	2	3	4
	Ohne Armbewegung	Mit Armbewegung	1 kg/Arm	2 kg/Arm
Romana	1,21m	1,47m	1,48m	1,32m
Thomas	1,71m	2,23m	2,29m	2,23m
Elias	1,18m	1,50m	1,70m	2,00m
Philipp	1,19m	1,55m	1,63m	1,73m
Theo	1,30m	1,68m	1,65m	1,40m
Claudia	1,11m	1,35m	1,30m	1,24m

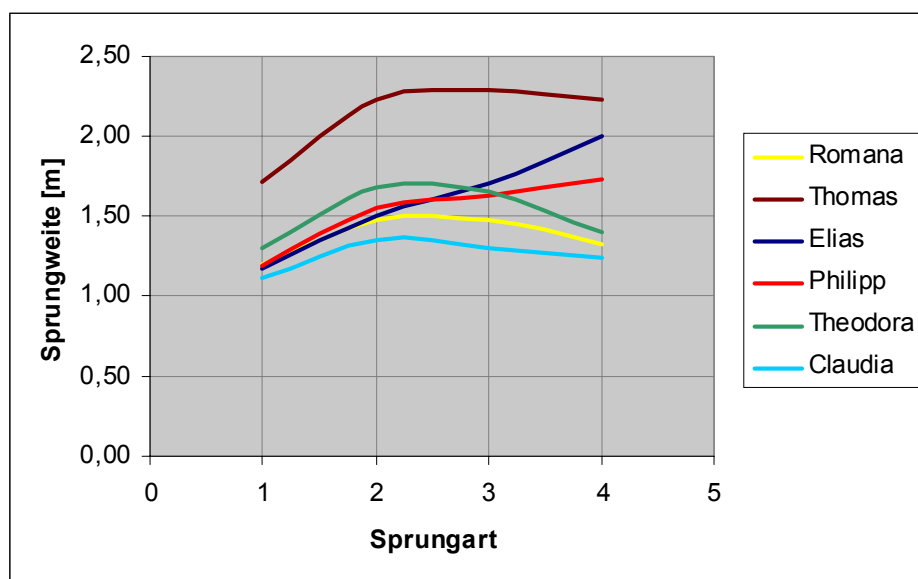


Abb. 14: Vergleich der Sprungweiten bei unterschiedlichen Sprungarten



Abb. 15: Sprungsequenz eines Standweitsprungs mit Armbewegung

Erhöhung der Sprungweite durch Wegwerfen der Halteres im Laufe des Sprungs

In manchen Quellen wird behauptet, dass die alten Griechen, um ihre Sprungweite zu erhöhen, im Laufe des Sprungs die Halteres nach hinten warfen. Wir stellten zu dieser Behauptung folgende Überlegungen an:

Durch das nach hinten Werfen der Gewichte bekommt der Springer einen zusätzlichen Impuls. Dieser erhöht den Eigenimpuls des Springers und somit auch seine Geschwindigkeit. Der Ort, an dem die Gewichte weggeworfen werden, ist leider nicht genau feststellbar. Ideal wäre es natürlich, wenn man sie gleich zu Beginn von sich stoßen würde. Da dies allerdings nicht möglich ist, nahmen wir an, dass dies nach $\frac{3}{4}$ der Strecke der Fall ist. Rechnerisch wurde nachgewiesen, dass das Wegwerfen der Gewichte an diesem Punkt eine Erhöhung der Sprungweite von ca. 2 cm je nach Weite ($=s_+$) bewirkt. Dies ist allerdings nur ein Richtwert, da die Sprungweite auch stark von den jeweiligen Muskeleigenschaften des Springers abhängt, die wir leider nicht messen konnten. Die Dauer des Sprunges wird durch den zusätzlichen Impuls aber nicht geändert.

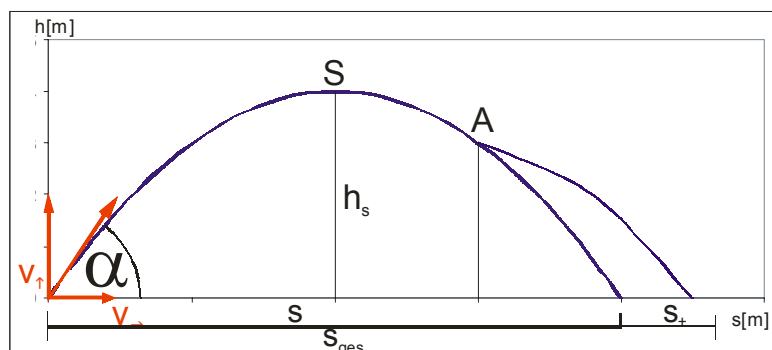


Abb. 16

Ist es sinnvoll, die Halteres während des Sprunges nach hinten zu werfen?

Wir verwendeten die Werte eines Sprunges unseres Probanden mit 2 kg Gewichten in jeder Hand, um zu errechnen, um wie viel sich seine Sprungweite erhöht, wenn er die Gewichte nach drei Vierteln der Zeit nach hinten wegwirft.

Gemessene Werte

$s = 2,23$
 $t = 0,65$
 $h = 0,3m$
 $h_1 = 1,2m$
 $h_2 = 1,5m$
 $m_{Hanteln} = 4kg$
 $m_{Person} = 69kg$
 $\alpha = 45^\circ$
 $x = 1,5m$



Abb.17

In einer Zeit von 0,65 Sekunden sprang unser Proband 2,23 Meter. Der gemessene Absprungwinkel betrug 45° .

Die Hanteln warf er aus einer Höhe von 1,2 Meter weg, wobei die Hanteln zuerst 0,3 Meter nach oben und dann erst zu Boden fielen. Die Flugweite der Hanteln betrug 1,5 Meter.

$t_1 = \sqrt{\frac{2h}{g}}$ $t_1 = 0,25s$ $t_2 = \sqrt{\frac{2h_2}{g}}$ $t_2 = 0,55s$ $t_{ges} = t_1 + t_2$ $t_{ges} = 0,80s$ $v_h = \frac{x}{t_{ges}}$ $v_h = 1,87m / s$	Rechenschritte: 1. Zeit, die die Gewichte nach dem Abwurf bis zum höchsten Punkt benötigen: t_1 2. Zeit vom höchsten Punkt bis zum Boden: t_2 3. Gesamtzeit vom Abwurf bis zum Bodenkontakt: t_{ges} 4. waagrechte Geschwindigkeit der Gewichte: v_h
$h_s = \frac{1}{2} g \left(\frac{t}{2}\right)^2$ $h_s = 0,52m$ $v_{\uparrow} = \sqrt{2gh_s}$ $v_{\uparrow} = 3,19m / s$ $v_{\rightarrow} = \frac{s}{t}$ $v_{\rightarrow} = 3,43m / s$	5. höchster Punkt des Schwerpunkts in der Sprungparabel: h_s 6. vertikale Komponente der Geschwindigkeit: $v_{\uparrow}$ 7. horizontale Komponente der Geschwindigkeit: $v_{\rightarrow}$

$p = m_{Hante\ ln} v_h$ $\underline{p = 7,50Ns}$ $v_k = \frac{p}{m_{Person}}$ $\underline{v_k = 0,11m/s}$	8. Impuls, der durch den Abwurf erzeugt wird: p 9. Geschwindigkeit, die durch den Abwurf gewonnen wird: v_k
$v_{end} = v_{\rightarrow} + v_k$ $\underline{v_{end} = 3,54m/s}$ $t_{end} = t \cdot \frac{1}{4}$ $\underline{t_{end} = 0,16s}$	10. Geschwindigkeit im letzten Viertel der Zeit: v_{end} 11. letztes Viertel der Zeit: t_{end}
$s_{end} = v_{end} t_{end}$ $\underline{s_{end} = 0,58m}$ $s_{anf} = \frac{3}{4} s$ $\underline{s_{anf} = 1,67m}$ $s_{ges} = s_{anf} + s_{end}$ $\underline{s_{ges} = 2,25m}$ $\underline{s_+ = 0,02m}$	12. Weg im letzten Viertel der Zeit: s_{end} 13. Weg in den ersten drei Vierteln der Zeit: s_{anf} 14. gesamter Weg: s_{ges} 15. gewonnener Weg durch Abwurf: s_+ <u>Ergebnis:</u> Es ist sinnvoll, da durch den Abwurf nach drei Vierteln der Zeit ca. 2 cm gewonnen werden.

Überprüfen des gemessenen Absprungwinkels α

$$\tan \alpha = \frac{v_{\uparrow}}{v_{\rightarrow}}$$

$$\alpha = 42,92^\circ$$

Der gemessene und der errechnete Wert stimmen näherungsweise überein.



Abb. 18: Standweitsprung mit Wegwerfen der Gewichte

Ist es sinnvoll, Halteres zu verwenden und diese während des Sprunges nach hinten zu werfen?

Zu Beginn haben wir die Absprunggeschwindigkeit durch die Armbewegung mit Gewichten simuliert und mit Matlab berechnet und anschließend im letzten Viertel der Zeit die horizontale Geschwindigkeit durch das Wegwerfen der Gewichte erhöht.

$t = 0,7s$ $s = 2,35m$ $v_{\rightarrow} = \frac{s}{t}$ $v_{\rightarrow} = 3,36m/s$	Die Zeit und die Länge wurden bei einem Sprung mit Armbewegung unseres Probanden gemessen (Thomas Janisch). <u>Aufgabenstellung:</u> Um wie viel springt er bei einem Sprung mit Zusatzgewichten von 2 kg je Arm weiter?
$h = \frac{1}{2} g \left(\frac{t}{2}\right)^2$ $h = 0,6m$ $v_{\uparrow} = \sqrt{2gh}$ $v_{\uparrow} = 3,43m/s$ $\tan \alpha = \frac{v_{\uparrow}}{v_{\rightarrow}}$ $\tan \alpha = 45,64^{\circ}$	<u>Rechenschritte:</u> 1. horizontale Komponente der Geschwindigkeit: $v_{\rightarrow}$ 2. höchster Punkt der Sprungparabel: h 3. vertikale Komponente der Geschwindigkeit: $v_{\uparrow}$ 4. Absprungwinkel: α 5. Anfangsgeschwindigkeit: v
$v = \sqrt{v_{\uparrow}^2 + v_{\rightarrow}^2}$ $v = 4,80m/s$ $p = \frac{100A}{G}$ $p = 103,71\%$	6. Da wir die Muskeleigenschaften unseres Probanden nicht kennen, haben wir mit Hilfe von Matlab die Prozente p ausgerechnet, um die ein computermodellierter Mensch mit 2kg-Hanteln schneller abspringt als ohne. (Verfälschung, da die Absprunggeschwindigkeit von den Muskeleigenschaften abhängt)
$v_{neu} = v \cdot \frac{p}{100}$ $v_{neu} = 4,98m/s$ $v_{neu\rightarrow} = \cos \alpha \cdot v_{neu}$ $v_{neu\rightarrow} = 3,48m/s$ $v_{neu\uparrow} = \sin \alpha \cdot v_{neu}$ $v_{neu\uparrow} = 3,56m/s$	7. Erhöhung von v um die errechneten Prozent 8. neue horizontale Komponente der Geschwindigkeit: $v_{neu\rightarrow}$ 9. neue vertikale Komponente der Geschwindigkeit: $v_{neu\uparrow}$

$h_{neu} = \frac{v_{neu\uparrow}^2}{2g}$ $h_{neu} = 0,65m$ $\sqrt{\frac{2h_{neu}}{g}} = t_{neu}$ $t_{neu} = 0,73s$ $s_{neu} = v \cdot t$ $s_{neu} = 2,53m$ $s_{neu} - s = s_+$ $s_+ = 0,18m$	10. neuer höchster Punkt der Sprungparabel: h_{neu} 11. neue Dauer des Sprungs: t_{neu} 12. neue Weite des Sprungs: s_{neu} <u>Ergebnis:</u> Er springt mit Zusatzgewichten von 2 kg pro Arm 2,53m weit, d.h. die Weite wurde um 18 cm (s_+) vergrößert.
$h = 1,5m$ $s = 1,5m$ $m = 4kg$ $m_t = 69kg$ $t_1 = \sqrt{\frac{2h}{g}}$ $t_1 = 0,25s$ $t_2 = 0,55s$ $t_i = t_1 + t_2$ $t_i = 0,80s$ $v_h = \frac{s}{t_i}$ $v_h = 1,87m/s$	<u>Aufgabenstellung:</u> Um wie viel springt er bei einem Sprung mit Zusatzgewichten von 2 kg je Arm weiter, wenn er die Zusatzgewichte nach $\frac{3}{4}$ der Zeit wegwirft? Abwurfhöhe: h Wurfweite: s Masse der Gewichte: m Masse des Probanden: m_t <u>Rechenschritte:</u> 13. Zeit, die die Zusatzgewichte brauchen, bis sie am Boden aufkommen: t_i 14. Geschwindigkeit, die die Gewichte haben: v_h
$p = mv_h$ $p = 7,50Ns$ $v_k = \frac{p}{m_t}$ $v_k = 0,11m/s$	15. Impuls, den der Proband durch das Wegwerfen der Zusatzgewichte bekommt: p 16. Geschwindigkeit, die durch den Impuls gewonnen wird: v_k

$v_s = v_{neu \rightarrow} + v_k$ $v_s = 3,59 m/s$ $s_s = v_s \cdot t_{neu} \cdot \frac{1}{4}$ $s_s = 0,66 m$ $s_a = s_{neu} \cdot \frac{3}{4}$ $s_a = 1,90 m$ $s_a + s_s = s_{ges}$ $s_{ges} = 2,55 m$ $s_{1+} = s_{ges} - s_{neu}$ $s_{1+} = 0,02 m$	17. Geschwindigkeit, die er nach dem Wegwerfen im letzten Viertel der Zeit hat: v_s 18. Weg, den er im letzten Viertel der Zeit zurücklegt: s_s 19. Weg in den ersten drei Vierteln der Zeit: a_s 20. gesamter Weg: s_{ges} <u>Ergebnis:</u> Er springt mit Zusatzgewichten von 2 kg pro Arm, wenn er diese nach drei Vierteln der Zeit wegwirft, um ca. 2,27 cm (s_{1+}) weiter.
--	---

Zusammenfassung:

Anhand unserer Simulationen sieht man, dass die Auswirkungen der Halteres für jede Person unterschiedlich sind. Ihre Effekte hängen von den individuellen Muskeleigenschaften, vom Körperbau, von der Armbewegung, von der Masse, usw. ab. Deshalb kann man keinen allgemein gültigen Wert für das optimale Gewicht der Halteres festlegen.

Quellenverzeichnis:

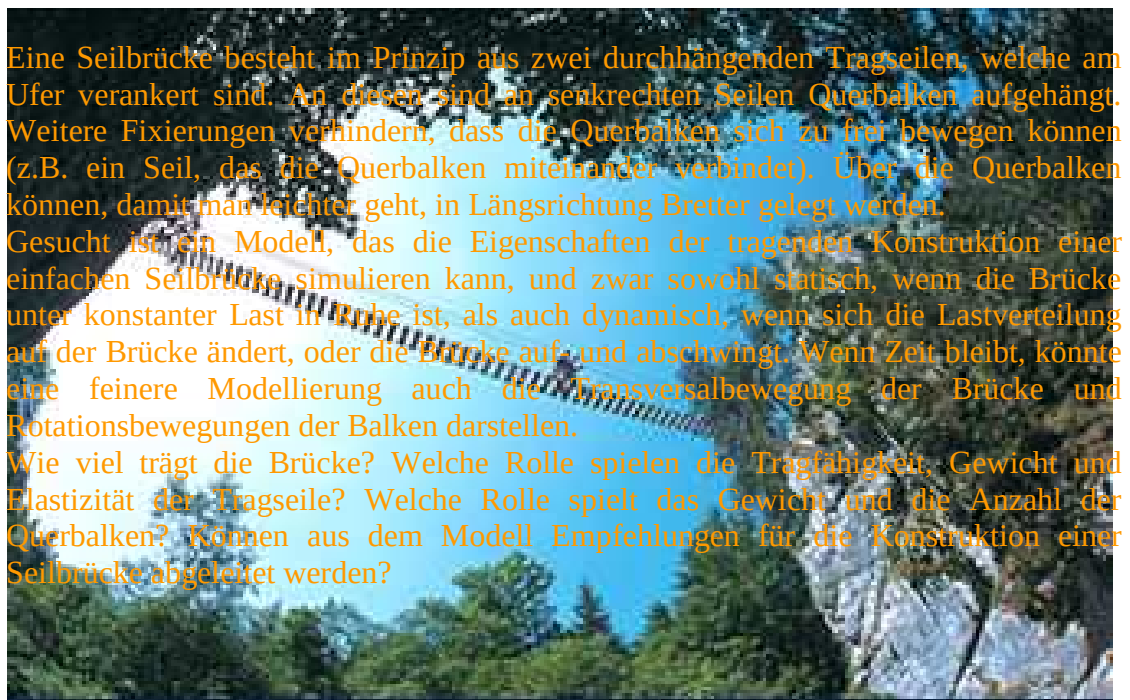
- [1] Alberto E. Minetti, Luca P. Ardigo (2002): Halteres used in ancient Olympic long jump; Nature 420: 141-142.
- [2] Matthieu Lenoir, Dirk de Clercq, & Willy Laporte (2005): The „how“ and „why“ of the ancient Greek long jump with weights: A five-fold symmetric jump in a row; Journal of Sports Sciences 23(10): 1033-1043.
- [3] <http://www.uni-graz.at/~sthaller/Model.htm> (Zugriff: 13.01.2006).
- [4] <http://www.uni-graz.at/~sthaller/muskelapplet.html> (Zugriff: 13.01.2006).

Dokumentation der Woche der mathematischen Modellierung

Arbeitsgruppe Mechanik unter der Leitung von Prof. Desch
Rita Ambach, Iris Bardel, Matthias Haselmann, Caroline Neuray, Hannes Plank, Stefan
Sorger

Modellierung einer SEILBRÜCKE

DIE PROBLEMSTELLUNG



DIE LÖSUNG

Ein Balken in 2D

Das erste Problem, mit dem wir uns bei der Modellierung unserer Seilbrücke auseinander setzen mussten, war der Zusammenhang zwischen den Größen, die auf diese Wirken.

Dazu haben wir ein vereinfachtes Modell einer Brücke herangezogen, das aus einer Masse, die einen Querbalken einer Brücke darstellt, und aus zwei Federn, die die Seile, an denen der Querbalken hängt, besteht.

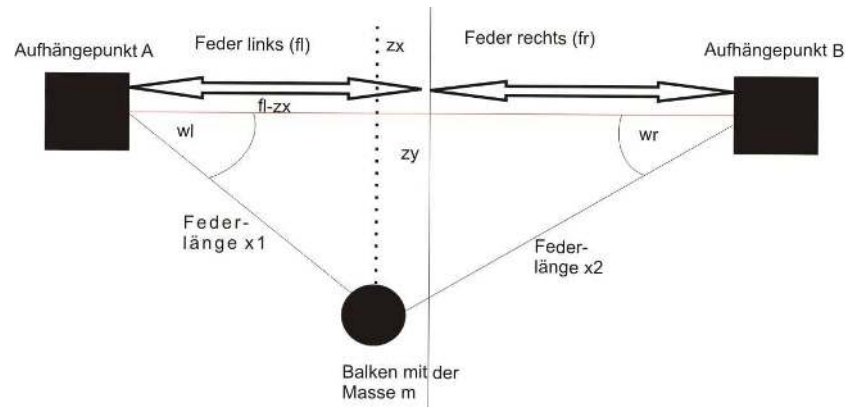
In diesem Modell haben wir uns die wichtigsten Begriffe und Zusammenhänge überlegt, mit Hilfe derer wir dann die Daten für die Brücke errechnen konnten. Dabei haben wir zwei Lösungswege entdeckt. Der erste ist uns durch die Winkel und Winkelfunktionen gelungen, die man von einem Punkt berechnen kann.

$$F_{y1} = k_{x1} \sin(wl)$$

$$F_{y2} = -k_{x2} \sin(wr)$$

$$F_{x1} = k_{x1} \cos(wl)$$

$$F_{x2} = k_{x2} \cos(wr)$$



$$\sum F_x = m a_x$$

$$-F_{x1} + F_{x2} = m a_x / m$$

$$a_x = -\frac{F_{x1}}{m} + \frac{F_{x2}}{m}$$

$$\sum F_y = m a_y$$

$$F_{y1} + (-F_{y2}) - mg = m a_y$$

$$a_y = \frac{F_{y1}}{m} - \frac{F_{y2}}{m} - g$$

$$\tan(wl) = \frac{zy}{fl - zx} \Rightarrow wl = a \tan \frac{zy}{fl - zx}$$

$$\tan(wr) = \frac{zy}{fr - zx} \Rightarrow wr = a \tan \frac{zy}{fr - zx}$$

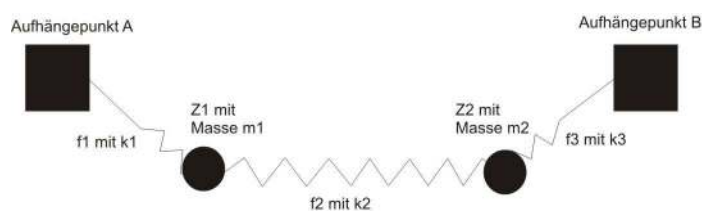
$$x_1 = \sqrt{(fl - zx)^2 + zy^2}$$

$$x_2 = \sqrt{(fr - zx)^2 + zy^2}$$

Zwei Balken in 2D

Als wir uns demselben Problem mit zwei Balken zugewandt haben, haben wir allerdings festgestellt, dass der Lösungsweg mit Vektoren um einiges weniger aufwendig und kompliziert ist.

Für diese Rechnung haben wir uns überlegt, welche Kräfte auf die einzelnen Balken wirken.



Auf den ersten Massepunkt wirken die Kräfte der beiden Federn, die nach links und rechts oben ziehen (f_1, f_2) und die Schwerkraft

$$fg = \begin{pmatrix} 0 \\ -g \end{pmatrix} m$$

Die Formel für die Kraft der ersten Feder lautet: $f_1 = k_1(\bar{A} - \bar{Z})$

Die Formel für die Kraft der zweiten Feder lautet: $f_2 = k_2(\bar{Z}_1 - \bar{Z}_2)$

Auf den zweiten Massepunkt wirkt die negative Kraft f_2 , die Schwerkraft und die Kraft der dritten Feder.

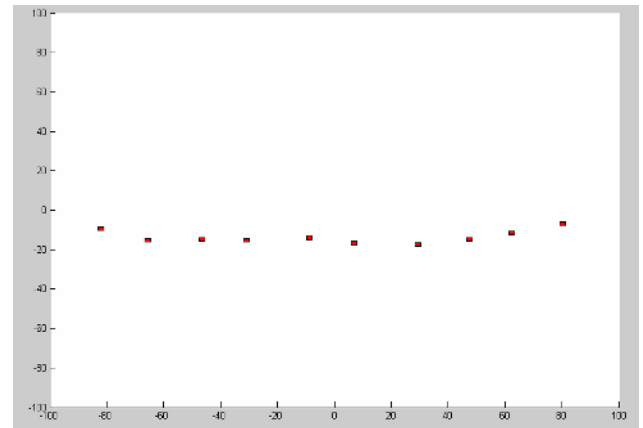
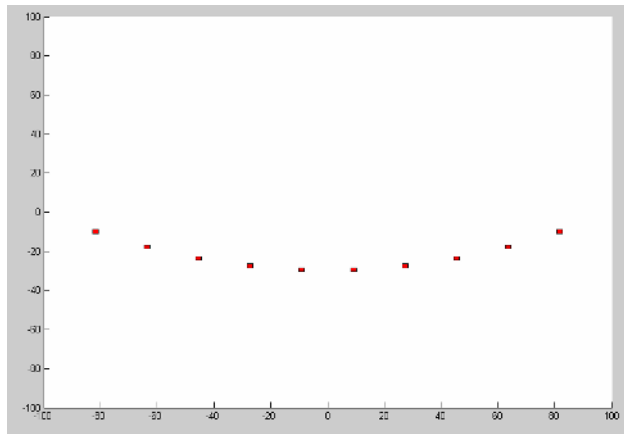
Die Formel für die Kraft der dritten Feder lautet: $f_3 = k_3(\bar{B} - \bar{Z}_2)$

Aus diesen Formeln kann man sich die Beschleunigung der Körper ausrechnen, bei der sowohl die Richtung, als auch die Stärke der Kraft miteinbezogen sind.

$$a_1 = \frac{1}{m_1} * (f_2 + f_1 + fg_2)$$

$$a2 = \frac{1}{m2} * (f3 - f2 + fg2)$$

Wenn man mit diesen Formeln das Bild im MATLAB programmiert, erhält man bereits das erste 2-dimensionale Modell einer Seilbrücke:

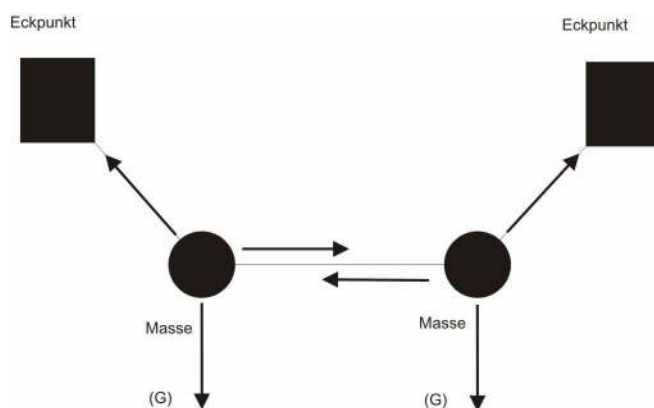


2D - Brücke in Ruhelage

Nachdem das Programm bereits dynamisch funktioniert hat, haben wir uns rechnerisch mit dem Problem der Brücke, wenn sie sich in 2D in Ruhelage befindet, auseinandergesetzt.

In Ruhelage heben sich alle Kräfte auf. Die Gesamtheit der Kräfte, die auf einen Balken wirken, muss null ergeben. Gesucht sind die Koordinaten $\begin{pmatrix} x \\ y \end{pmatrix}$ vom Balken. Alle anderen

Werte sind bekannt.



Auf Grund dieser Voraussetzung haben wir folgende Formel gefunden, die vorerst nur für einen Balken gilt.

$$\begin{pmatrix} k_{1a} - k_{1x} \\ k_{1b} - k_{1y} \end{pmatrix} + \begin{pmatrix} k_{2c} - k_{2x} \\ k_{2d} - k_{2y} \end{pmatrix} + \begin{pmatrix} 0 \\ -gm \end{pmatrix} = 0$$

Diese Formel haben wir anschließend in zwei Teile zerlegt, damit wir für die zwei Unbekannten x und y zwei Formeln haben.

- I. $k_{1a} - k_{1x} + k_{2c} - k_{2x} = 0$
- II. $k_{1b} - k_{1y} + k_{2d} - k_{2y} - gm = 0$

Bei jeder der beiden Formeln haben wir jetzt nur mehr eine Unbekannte und können beide durch uns bekannte Zahlen ausdrücken.

$$x = \frac{k_1 a - k_2 c}{k_1 + k_2}$$

$$y = \frac{k_1 b + k_2 d - g m}{k_1 + k_2}$$

Als nächstes haben wir angenommen, dass die Brücke zwei Querbalken hat und dafür die Formeln gefunden. → Skizze?

$$\begin{pmatrix} k_1 a - k_1 x \\ k_1 b - k_1 y \end{pmatrix} + \begin{pmatrix} k_1 w - k_2 x \\ k_2 z - k_2 y \end{pmatrix} + \begin{pmatrix} 0 \\ -g m_1 \end{pmatrix} = 0$$

$$\begin{pmatrix} k_2 x - k_2 w \\ k_2 y - k_2 z \end{pmatrix} + \begin{pmatrix} k_3 c - k_3 w \\ k_3 d - k_3 z \end{pmatrix} + \begin{pmatrix} 0 \\ -g m_2 \end{pmatrix} = 0$$

Auch diese Formeln haben wir in vier Teile geteilt.

- I. $k_1 a - k_1 x + k_2 w + k_2 x = 0$
- II. $k_1 b - k_1 y + k_2 z - k_2 y - g m_1 = 0$
- III. $k_2 x - k_2 w + k_3 c - k_3 w = 0$
- IV. $k_2 y - k_2 z + k_3 d + k_3 z - g m_2 = 0$

Würden wir diese vier Formeln auf die Weise umformen, wie wir es mit den zwei Formeln gemacht haben, bekämen wir lange und komplizierte Rechnungen. Deshalb haben wir eine Methode kennen gelernt, mit der das Umformen einfacher geht.

x	w	y	z	=
-k ₁ -k ₂	k ₂	0	0	k ₁ a
0	0	-k ₁ -k ₂	k ₂	-k ₁ b g m ₁
k ₂	-k ₂ -k ₃	0	0	-k ₃ c
0	0	k ₂	-k ₂ -k ₃	-k ₃ d + g m ₂

Es gibt fünf Spalten, die für die vier Unbekannten und für die Terme, in denen keine Unbekannten vorkommen, stehen. Die Zeilen stehen für die vier Formeln. In jeder der Zeilen stehen unter den Unbekannten die Faktoren, die in vor den Unbekannten stehen. In der fünften Spalte steht der Rest, wobei die Vorzeichen umgedreht werden, da die Faktoren auf die andere Seite gebracht werden.

Diese Formeln lassen sich auf Grund ihrer Gleichheit der Formeln in zwei Teile teilen: eine Tabelle mit x und w, eine Tabelle mit y und z, die anschließend wieder extra angeschrieben werden.

Im Endzustand sollten in jeder Spalte mit Unbekannten eine 1 und sonst nur 0 stehen, alle Zahlen sollten in der Spalte mit = stehen. Das wird durch Umformen erreicht.

x	w	=
0	1	$\frac{k_1 k_2 a + k_1 k_3 c + k_2 k_3 c}{k_1 k_2 + k_2 k_3 + k_1 k_2}$

1	0	$\frac{a(k_1k_2 + k_1k_2k_2) + c(k_2k_3)}{k_1k_2 + k_2k_3 + k_1k_3}$
---	---	--

Die Lösungen für die Unbekannten stehen unter dem =. Sie gelten immer für die Unbekannte, unter der die 1 steht.

Dasselbe haben wir auch für y und z gemacht.

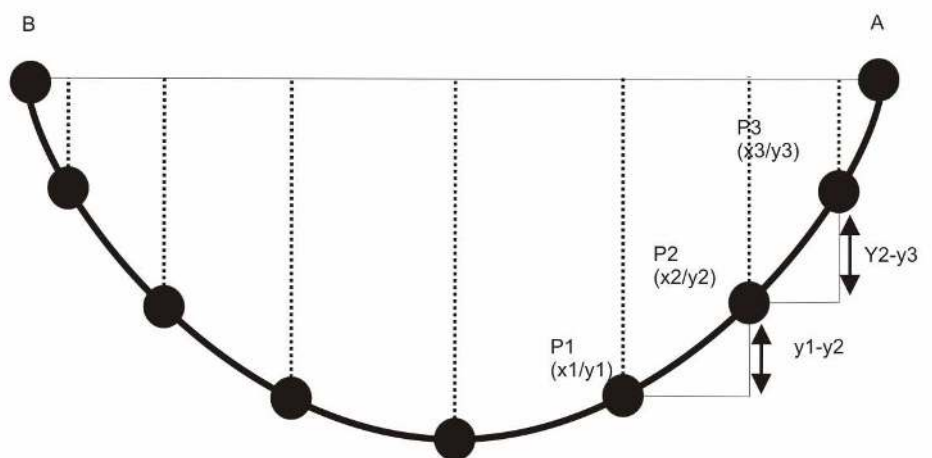
Um das System hinter dieser Methode herauszufinden, haben wir dasselbe noch einmal mit vier Querbalken, also acht Unbekannten versucht.

x_1	y_1	x_2	y_2	x_3	y_3	x_4	y_4	=
$-k_1-k_2$	0	k_2	0	0	0	0	0	$-k_1a$
0	$-k_1-k_2$	0	k_2	0	0	0	0	$-k_1b+g*m_1$
k_2	0	$-k_2-k_3$	0	k_3	0	0	0	0
0	k_2		$-k_2-k_3$	0	k_3	0	0	$g*m_2$
0	0	k_3	0	$-k_3-k_4$	0	k_4	0	0
0	0	0	k_3	0	$-k_3-k_4$	0	k_4	$g*m_3$
0	0	0	0	k_4	0	$-k_4-k_5$	0	$-k_5c$
0	0	0	0	0	k_4	0	$-k_4-k_5$	$-k_5d+g*m_4$

Durch diese Tabelle konnten wir das System herausfinden und die Rechnung leicht auch für mehr Balken wiederholen, was vor allem fürs Programmieren von Vorteil war.

Beim Programmieren und der Berechnung der statischen Brücke ist uns aufgefallen, dass, wenn alle k-Werte und alle m-Werte gleich sind, die x-Werte immer im gleichen Abstand und die y-Werte in ganzzahligen Verhältnissen zueinander liegen.

Das haben wir anschließend mathematisch erklärt.



I. $k(x_{i-1} - x_i) + k(x_{i+1} - x_i) = 0$

II. $k(y_{i-1} - y_i) + k(y_{i+1} - y_i) = 0$

I. $k(x_{i-1} - x_i) = k(x_i - x_{i+1})$

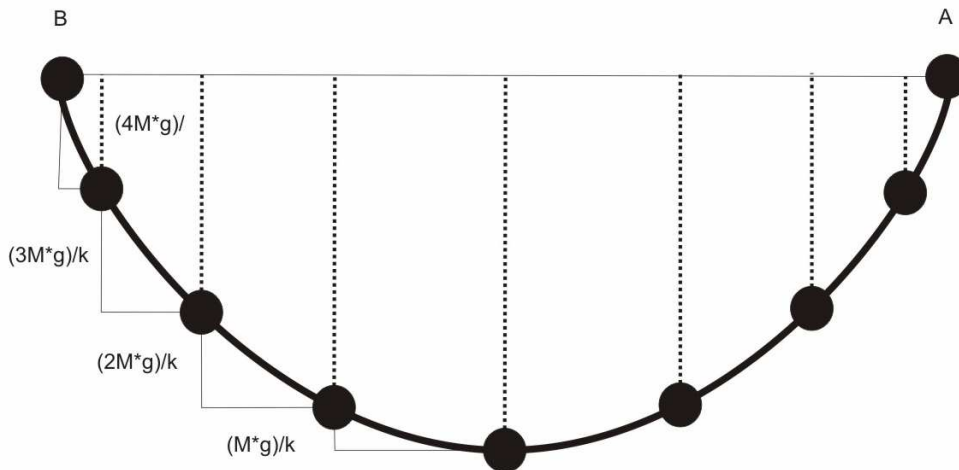
$x_{i-1} - x_i = x_i - x_{i+1}$

Bei diesen Formeln sieht man, dass alle x-Werte den gleichen Abstand zueinander haben. Das ergibt sich, weil auf jeden Punkt die gleichen Kräfte wirken, die ausgeglichen sind.

$$k(y_{i-1} - y_i) = k(y_i - y_{i+1}) + \frac{mg}{k}$$

$$\text{II. } y_{i-1} - y_i = y_i - y_{i+1} + \frac{mg}{k}$$

$$y_i - y_{i-1} = y_{i+1} - y_i - \frac{mg}{k}$$



Aus dieser Formel geht hervor, dass sich die Abstände zwischen den y-Werten nur um $\frac{mg}{k}$ unterscheiden. Deshalb stehen sie immer in ganzzahligen Verhältnissen zueinander.

Das Problem eines Balkens in 3D

Um das Modell zu vereinfachen, haben wir als erstes die Masse auf die zwei Eckpunkte verteilt. Die Eckpunkte heißen E1 und E2. Um ihre Positionen feststellen zu können, müssen wir alle wirkenden Kräfte in Form von Vektoren addieren. Auf jeden der beiden Eckpunkte wirken vier Kräfte. Die

der Seile, die die Eckpunkte in Richtung der Aufhängepunkte ziehen, die Schwerkraft und eine uns unbekannte Kraft, die verhindert, dass die Eckpunkte auseinander wandern. Sie übernimmt die Aufgabe des Stabs (wir haben ja die Masse aus rechnerischen Gründen auf die Eckpunkte konzentriert).

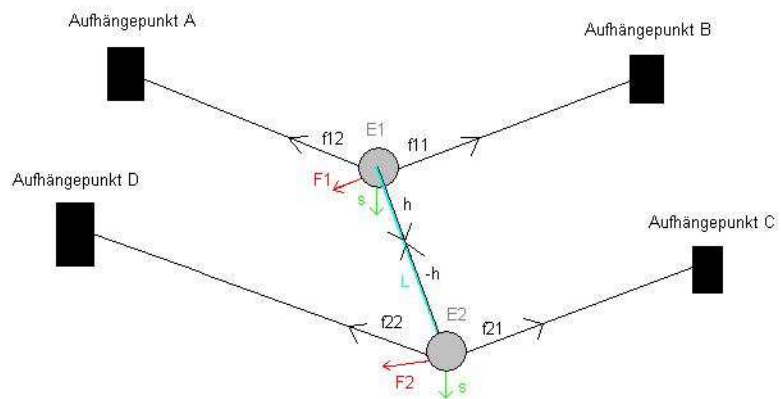
Was wir sicher wissen ist, dass der Abstand zwischen den Eckpunkten immer gleich ist (der Stab ändert seine Länge nie). Das bedeutet:

$$|\vec{E1} - \vec{E2}| = \text{const}$$

$$L = (x1 - x2)^2 + (y1 - y2)^2 + (z1 - z2)^2 = \text{const}$$

$$\Delta L = 0$$

$$[(x1 - x2)^2 + (y1 - y2)^2 + (x1 - x2)^2] = 0$$



Jetzt haben wir nach einer Formel gesucht, die diese Formel mit der Beschleunigung in Verbindung bringt. Also haben wir die Formel abgeleitet. Die erste Ableitung gibt uns die Geschwindigkeit und die zweite Ableitung die Beschleunigung. Von der zweiten Ableitung haben wir das skalare Produkt gebildet (wir werden es nach weiteren Rechenschritten wieder brauchen):

$$\text{Skalarprodukt der 2. Ableitung: } (E1 - E2) \bullet (E1'' - E2'') + (E1' - E2') \bullet (E1' - E2')$$

Wir wissen aber noch immer nicht, wie stark die Kraft, die die Eckpunkte zusammenhält (=h), ist. Diese unbekannte Stärke nennen wir γ .

$$h = \gamma(E1 - E2)$$

Hier kommt die Beschleunigung ins Spiel.

$$a = \frac{F}{m} \quad F = F1,2 + h$$

für E1: $F1 = \text{Summe aller Kräfte, die auf E1 wirken}$

$$a = \frac{2}{m} \{ F1 + [\gamma(E1 - E2)] \} \rightarrow \text{Beschleunigung von E1}$$

für E2: $F2 = \text{Summe aller Kräfte, die auf E2 wirken}$

$$a = \frac{2}{m} \{ F2 - [\gamma(E1 - E2)] \} \rightarrow \text{Beschleunigung von E2}$$

Bis jetzt γ ist noch immer unbekannt. Um es auszudrücken müssen wir es in das skalare Produkt der zweiten Ableitung einsetzen. Das gibt folgende Gleichung:

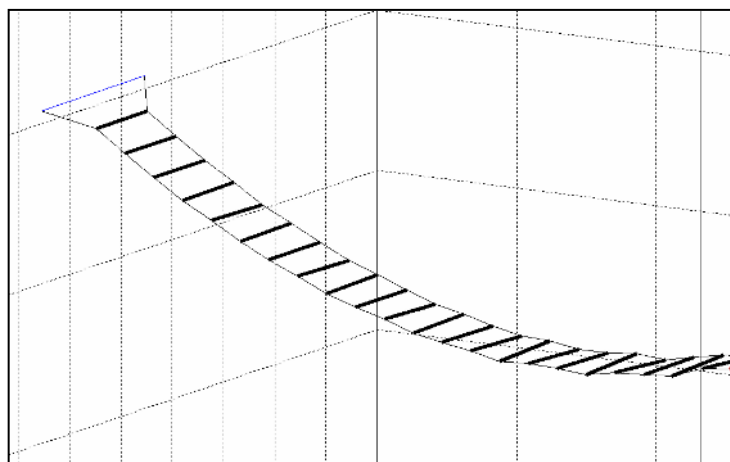
$$\begin{aligned} & \frac{2}{m} (E1 - E2) \bullet F1 + \frac{2}{m} \gamma (E1 - E2) \bullet (E1 - E2) - \frac{2}{m} (E1 - E2) \bullet F2 + \\ & + \frac{2}{m} \gamma (E1 - E2) \bullet (E1 - E2) + (E1' - E2') \bullet (E1' - E2') = 0 \end{aligned}$$

Um zu vereinfachen und γ auszudrücken:

$$\begin{aligned} L^2 &= (E1 - E2) \bullet (E1 - E2) \\ \gamma &= - \frac{(E1 - E2) \bullet F1}{2L^2} + \frac{(E1 - E2) \bullet F2}{2L^2} - \frac{(E1' - E2') \bullet (E1' - E2')}{2 + \frac{2}{m} L^2} \end{aligned}$$

Sobald wir γ ausgedrückt haben, sind uns alle Faktoren bekannt und wir haben eine Formel, die alles über den Balken in 3D aussagt und somit können wir eine erste 3D Brücke programmieren.

Allgemein ist es das Ziel dem Balken so viel Freiheit wie möglich zu lassen. Deshalb haben wir im nächsten Schritt versucht, waagrechte Drehung, Kippdrehung, Trägheitsmoment und Drehmoment mit einzubeziehen. Dafür muss man weitere Faktoren erklären:



φ ... Drehwinkel der waagrechten Drehung
 ψ ... Drehwinkel der Kippdrehung
 u ... Position des Schwerpunktes

F ... Summe aller angreifender Kräfte

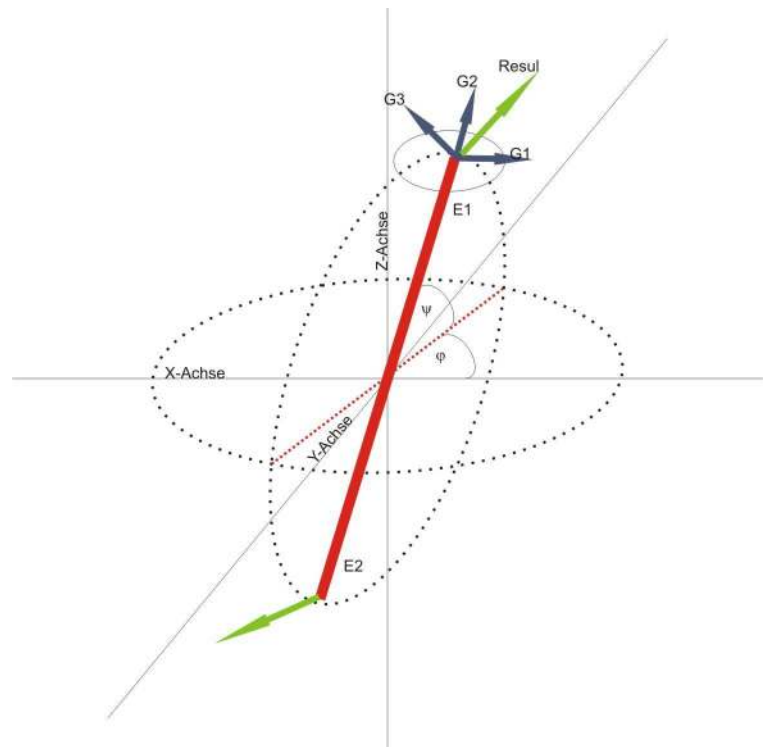
$$F = ma \quad a = u''$$

$$^1F = mu'' \quad F = f_1 + f_2 + mg$$

Drehimpuls = Trägheitsmoment * Winkelgeschwindigkeit (erste Ableitung)

Δ Drehimpuls = Drehmoment

$$T = \frac{ml^2}{12}$$

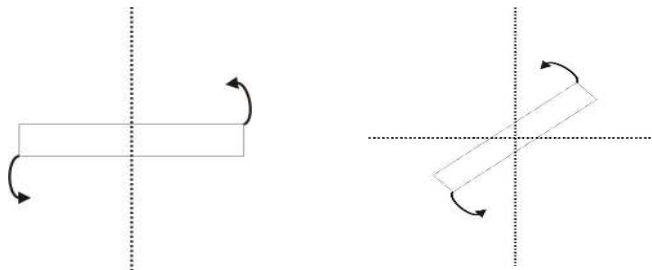


T ... Trägheitsmoment wenn sich der Balken quer zur seiner Längsachse dreht

Trägheitsmoment: $T \cos^2(\psi)$

→ Je näher ψ an der senkrechten Position ist, desto kleiner wird der Trägheitsmoment weil der Abstand zur Drehachse kleiner ist

→ wenn $\cos = 90$, ist die Trägheit null, weil der Balken senkrecht steht



Drehmoment = Kraft * Kraftarm → bestimmt, wie stark sich der Balken dreht

Änderung des Drehimpuls von ψ :

Die Änderung des Drehimpulses der Kippdrehung erfolgt auf Grund der Drehmomente durch die Seilkräfte und der Zentrifugalkraft der waagrechten Drehung.

$$T \psi'' = G_3 - (T \cos \psi \sin \psi \varphi'^2)$$

$(T \cos \psi \sin \psi \varphi) =$ Zentrifugalkraft → versucht ψ zu verkleinern

Änderung des Drehimpuls von φ :

¹ Herleitung mit Hilfe des Lagrangeprinzips

Die Änderung der waagrechten Drehung: Der Drehimpuls der waagrechten Drehung hängt vom Trägheitsmoment und von der Winkelgeschwindigkeit ab. Weil sich das Trägheitsmoment mit ψ ändert, tritt auch hier eine Wechselwirkung zwischen ψ und φ auf.

$$\cos(\psi) T \varphi'' = G_1 \cos \psi + 2T \varphi' \psi' \sin \psi$$

G1...Drehmoment, der auf φ wirkt $\rightarrow$ versucht φ zu ändern

G2...hat keine Auswirkung

G3...Drehmoment, der auf ψ wirkt $\rightarrow$ versucht ψ zu ändern

G...Gesamtdrehmoment

$$G = \frac{l}{2} f_1 - \frac{l}{2} f_2$$

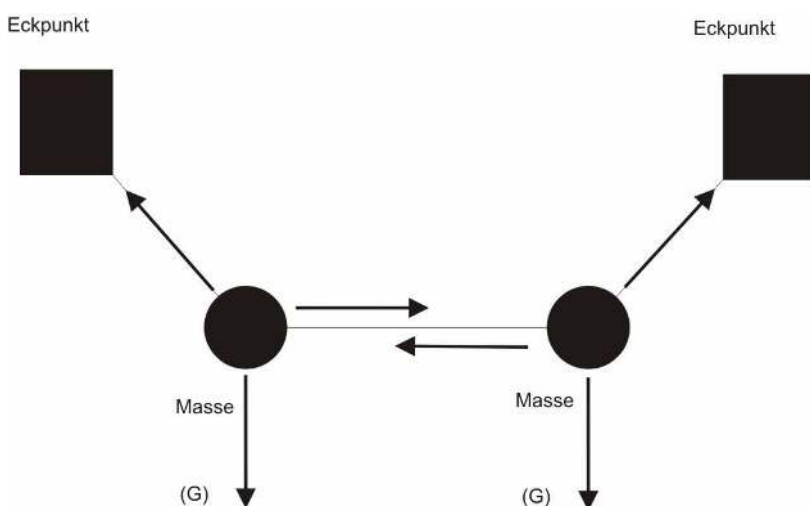
f1... Seilkräfte an E1

f2... Seilkräfte an E2

$$G_1 = G \cdot \begin{pmatrix} -\cos \varphi \\ -\sin \varphi \\ 0 \end{pmatrix}$$

$$G_3 = G \cdot \begin{pmatrix} \sin \psi \sin \varphi \\ -\sin \psi \cos \varphi \\ \cos \psi \end{pmatrix}$$

Das statische Problem



Um die Brücke möglichst naturgetreu darzustellen, muss man berücksichtigen, dass Seile ja keine Federn sind. Federn würden wenn sie rissen auf null zurückstellen was Seile nicht tun da sie durchhängen können. Um diese Verbesserung zu simulieren, haben wir angenommen, dass vor der Feder ein unelastisches ungespanntes Stück Seil ist. Von diesem

Seil kennen wir die Länge u. Wir wollen berechnen, wo sich der Massepunkt (= Balken) befindet, wenn das Seil gespannt ist und die Federn in Ruhe sind.

Sobald das Seil gespannt ist, wirken drei Kräfte – f1, die von P zu dem unbekannten Punkt C zieht, f2, die von P zum Aufhängepunkt B zieht und die Schwerkraft s, die die Masse nach unten zieht. Gesucht ist die Ruhelage, wofür wir den Punkt C ausrechnen müssen.

Dafür nehmen wir an, dass uns die Koordinaten des Punktes C bekannt sind. Wir wissen, dass der Punkt C die gleiche Richtung hat wie der Vektor AP. Zur Berechnung der Länge nehmen wir den Einheitsvektor von AX und multiplizieren ihn mit u. Das ergibt Folgendes:

$$AC = \frac{u}{|AX|} AX$$

$$XC : XA + AC = XA - \frac{u}{|XA|} XA$$

$$|XA| = n$$

$$XC = \left(1 - \frac{u}{n}\right) XA \rightarrow \text{davon sind uns jetzt alle Vektoren bekannt}$$

Im nächsten Schritt müssen wir die Kräfte berechnen, denn eine Ruhelage ist dann gegeben, wenn alle wirkenden Kräfte aus null summieren.

Federkräfte: $f_1 = k XC$

$f_2 = k XB$

$$s = \begin{pmatrix} 0 \\ -mg \end{pmatrix} \quad f_1 + f_2 + s = 0$$

Nach einigen Berechnungen können wir x und y ausdrücken:

$$x = \frac{cn + an - au}{2n - u} \quad y = \frac{(n - u)b + nd - \frac{mng}{k}}{2n - u}$$

Das Problem, das sich jetzt ergibt ist, dass uns n noch immer unbekannt ist, und dass in n x und y enthalten sind. Das führt uns zum nächsten Schritt – nämlich n berechnen.

$$n = \sqrt{(a - x)^2 + (b - y)^2}$$

Erst kümmern wir uns um (a-x):

$$a - x = a - \frac{(n - u)a + cn}{2n - u} \rightarrow \frac{n}{2n - u} (a - c)$$

Teilschritt (b-y):

$$b - y = b - \frac{(n - u)b + dn - \frac{mgn}{k}}{2n - u} \rightarrow \frac{n}{2n - u} \left(b - d + \frac{gm}{k} \right)$$

Wenn man das jetzt in die Formel von n einsetzt ergibt das nach Berechnung Folgendes:

$$n = \frac{u + \left(\sqrt{(a - c)^2 + \left(d - b + \frac{mg}{k} \right)^2} \right)}{2}$$

$$XA = \begin{pmatrix} a - x \\ b - y \end{pmatrix}, \quad XB = \begin{pmatrix} c - x \\ d - y \end{pmatrix}, \quad s = gm$$

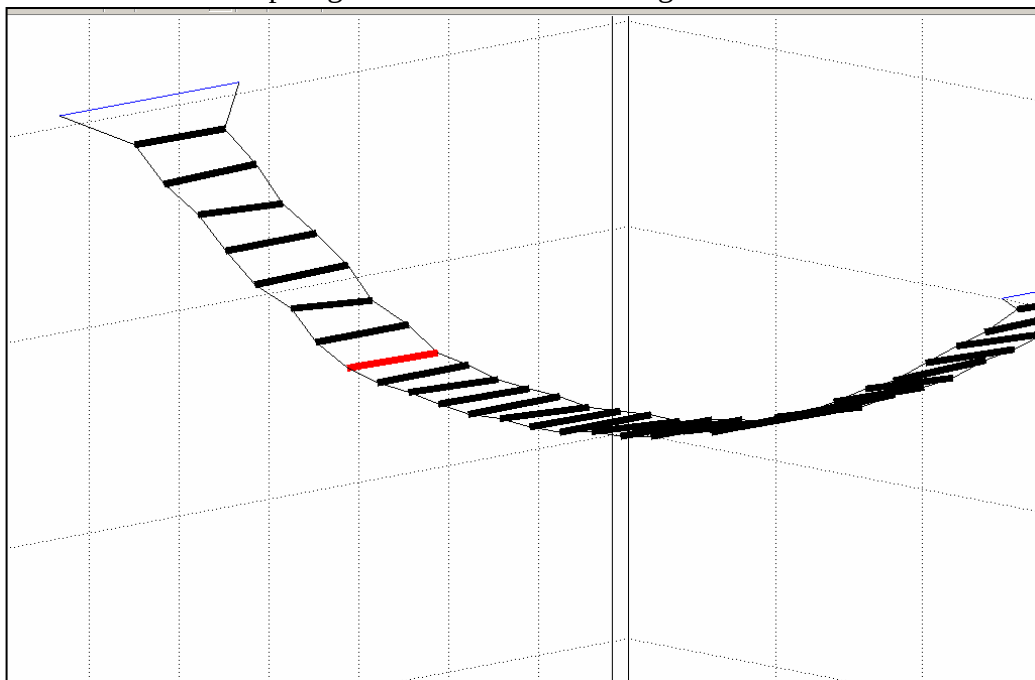
Sobald man n kennt, kann man alles zu einer Formel zusammenfügen und somit in das Programm einfügen

Programmetechnische Entwicklung

Um den Grundstein für die Programmierung zu legen, bekamen wir gleich am ersten Tag eine Einführung in das Programm Matlab, das uns die Simulation von Seilbrücken ermöglichen sollte. Am Abend konnten wir bereits die erste naturgetreue 2D-Simulation eines Massepunktes, welcher an zwei Federn befestigt ist, vorweisen. Am nächsten Morgen erweiterten wir diese Simulation auf zwei Massepunkte, die durch eine dritte Feder untereinander verbunden waren, was uns in gewisser Weise für die Simulation einer Seilbrücke vorbereiten sollte, welche wir am späten Nachmittag fertig stellten.

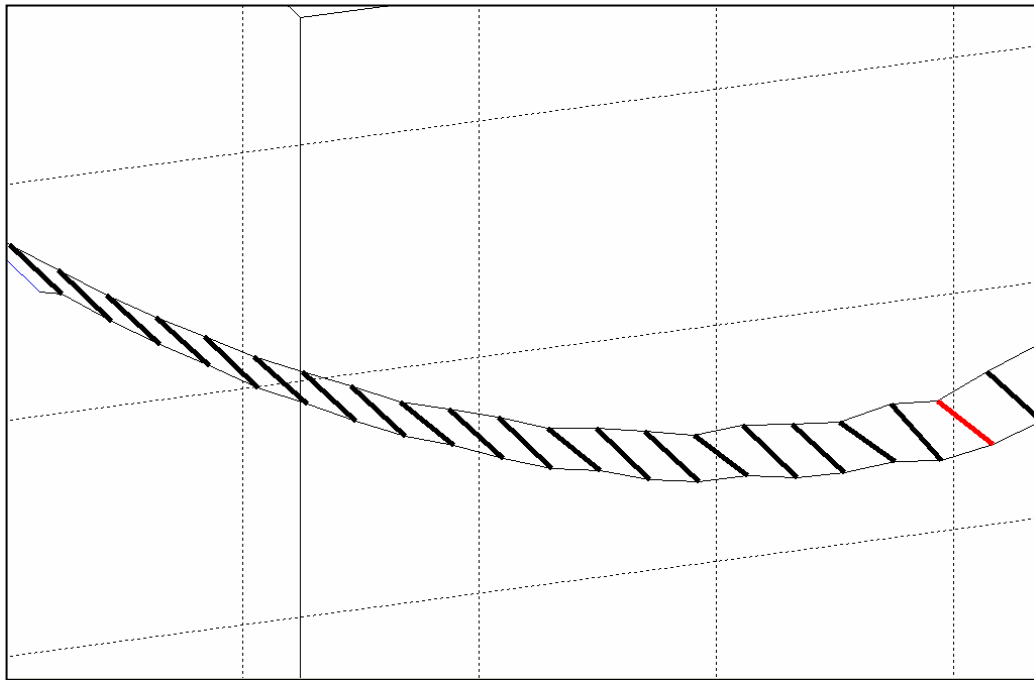
Der nächste Schritt war die Simulation einer 3D-Brücke, für die einige weiterführende Informationen in Matlab notwendig waren. Die Erweiterung auf drei Dimensionen erforderte wesentlich mehr Parameter, die berücksichtigt werden mussten. Deshalb brauchten wir einen ganzen Tag um die Brücke lediglich ohne Transversalbewegungen zu programmieren. Da wir aber in dieser Version alles dafür miteinbezogen haben, konnten wir unser Modell am nächsten Tag modifizieren, sodass wir auch die Transversalbewegungen bewundern konnten.

Als wir jedoch versuchten die bis jetzt verwendeten Federn, die die Brückenbalken verbunden hatten, durch Seile mit einer sehr hohen Federkonstante zu ersetzen, ergaben sich schwer zu lösende Probleme. Durch die hohen Federkonstanten der Seile, entstehen äußerst hohe Frequenzen, die mit der niedrigen Frequenz des gesamten Systems, also der Seilbrücke, eine steife Funktion kombiniert sind, sodass sich eine steife Differentialgleichung ergibt, welche mit unseren Methoden nur schwer auf akzeptable Weise zu lösen ist. Den bis dahin verwendeten Zeitschritt von einer Millisekunde mussten wir auf eine Mikrosekunde erniedrigen, um überhaupt Simulationen zu erhalten, die nicht nach mehreren Sekunden kollabieren. Das wäre an sich kein Problem gewesen, wenn die Rechenzeit nicht enorm vergrößert worden wäre. Ursprünglich konnten unsere Programme in den Simulationen alle

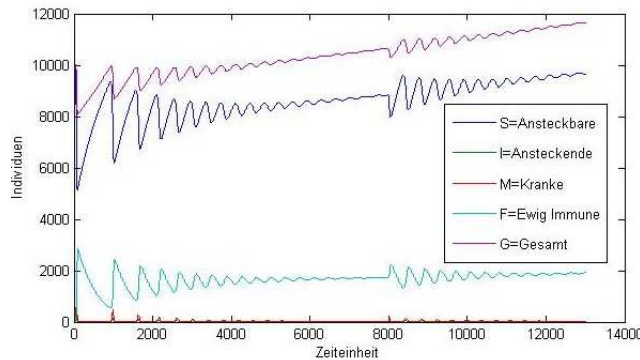


nötigen Daten in Echtzeit berechnen, während nach dieser Modifikation die Computer mehrere Stunden(!) rechnen mussten um die gespeicherten Daten nachher abspielen zu

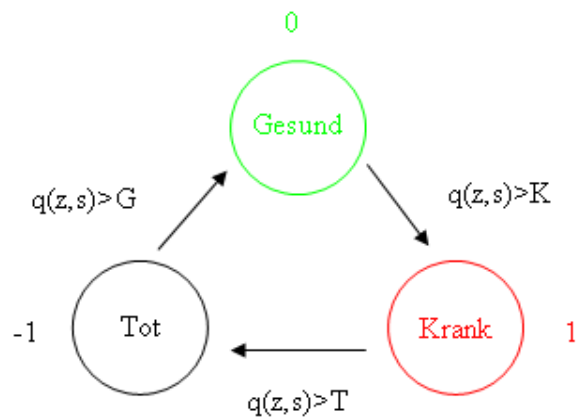
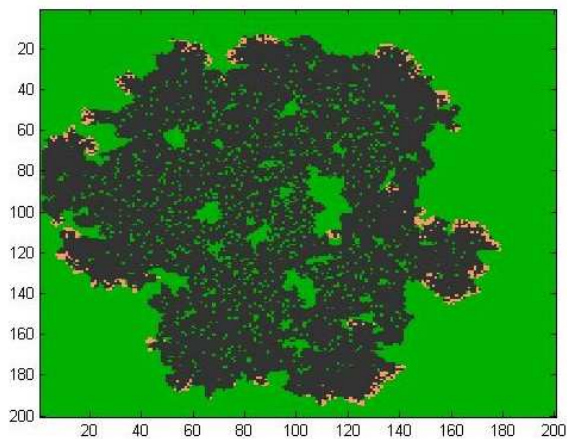
können. Dass wir unter diesen Umständen unsere neu modifizierte Simulation nicht ausreichend testen und verbessern konnten ist verständlich. Doch im Großen und Ganzen kam unsere simulierte Seilbrücke der Realität sehr nahe.



Biowissenschaften- und Sozialwissenschaften



Ausbreitung ansteckender Krankheiten



Dr. Georg Propst

Maria Barelli, Karlheinz Breinhalter, Florian Langmann, Martin Perz,
Cornelia Reinharter, Sofie Walzl

Themenbeschreibung

Infektiöse Krankheiten können durch den Kontakt von ansteckbaren und ansteckenden Individuen übertragen werden. Je nach dem Verhalten der Gruppe, den individuellen Eigenschaften der Krankheit und den Maßnahmen, die eingeleitet werden, verläuft die Ausbreitung unterschiedlich.

Modell 1

Hierbei wird gezeigt, welche Auswirkungen der Ausbruch einer ansteckenden Krankheit auf eine gut durchmischte Gruppe beweglicher Individuen (z.B.: Fischteich, Hühnerstall, Kindergarten, etc) hat.

Am Beginn besteht die Population aus einer großen, konstanten Gruppe (G), die in zwei Untergruppen, die gesunden Ansteckbaren (S - susceptible) und die infizierten Ansteckenden (I - infectious), unterteilt wird.

$$S_n + I_n = G$$

für alle $n = 1, 2, 3, \dots, n$

Solange für $S_n > 0$ und $S_n + I_n \leq G$ gilt, werden zwischen n und $(n+1)$ S_n angesteckt und werden zu I_n .

Man muss nun zwei Fälle unterscheiden:

1.) $S_n \geq I_n$

$$S_n \geq I_n \begin{cases} I_{n+1} = 2I_n \\ S_{n+1} = G - 2I_n = S_n - I_n \end{cases}$$

Beweis:

$$I_n + S_n = I_0 + S_0$$

$\Downarrow$

$$\begin{aligned} I_{n+1} + S_{n+1} &= 2I_n + S_n - I_n = \\ &= I_n + S_n = I_0 + S_0 \end{aligned}$$

2.) $S_n < I_n$

$$S_n < I_n \begin{cases} I_{n+1} = S_n + I_n \\ S_{n+1} = 0 \end{cases}$$

Beweis:

$$S_n + I_n = S_{n+1} + I_{n+1} = S_0 + I_0$$

Das Modell wird anschließend folgendermaßen verallgemeinert:

Jeder Infizierte steckt pro Zeiteinheit (zwischen n und $(n+1)$) x Ansteckbare an ($x > 0$).

Man unterscheidet auch hierbei zwei Fälle:

$$1.) S_n \geq x * I_n$$

$$2.) S_n < x * I_n$$

$$x * I_n \leq S_n \begin{cases} I_{n+1} = I_n + x * I_n = (1 + x) * I_n \\ S_{n+1} = S_n - x I_n \end{cases}$$

$$x * I_n > S_n \begin{cases} I_{n+1} = I_0 + S_0 \\ S_{n+1} = 0 \end{cases}$$

Die Überlegung ist, dass ein Individuum sich nicht selbst infizieren kann. Daher wird von der Gesamtheit, die infiziert werden kann, eins abgezogen.

$$I_{n+1} = x \frac{G - I_n}{G - 1} * I_n + I_n$$

Diese Überlegung ist zwar richtig, aber für nicht ganzzahlige Gruppengrößen ungeeignet, weil sonst die Möglichkeit besteht, dass $G < 1$ ist und so die Anzahl der Infizierten negativ wird.

Deshalb wird die Gesamtheit nicht um 1 verringert. Das hat zur Folge, dass die Gleichung im positiven Bereich bleibt, aber für kleine Gruppe unexakt wird.

Bei großen Gruppen kann dies vernachlässigt werden, da das abgezogene Individuum im Verhältnis zur gesamten Gruppe nicht ins Gewicht fällt.

Für den vereinfachten Term können wir die mögliche Negativität durch folgenden Beweis ausschließen:

$$S_n > 0 \wedge I_n \geq 0$$

$$\text{Annahme} \Rightarrow S_n + 1 > 0$$

$$S_n - z * \frac{S_n}{S_n + I_n} I_n < 0$$

$$S_n (S_n + I_n) < z I_n * S_n$$

$$S_n + z * I_n < 0$$

Dieser ist beim Einsetzen von positiven Werten sicher positiv. Damit ist die Annahme, dass der gegebene Term, nicht negativ werten kann, bewiesen.

Das x wird in weiterer Folge durch einen Term, der die Kontaktaufnahme darstellt, und den Parameter der Ansteckungswahrscheinlichkeit η (Eta, in der Physik der Wirkungsgrad), der die Aggressivität der Krankheit angibt, ersetzt.

$$I_{n+1} = \eta * \frac{S_n}{S_n + I_n} * I_n + I_n$$

$$S_{n+1} = S_n - \eta * \frac{S_n}{S_n + I_n} * I_n$$

Daraufhin werden als neue Klasse die *Morituri* (M) eingeführt. Bei diesen ist die Krankheit bereits ausgebrochen und sie befinden sich deshalb in Quarantäne, wo sie keine weiteren S_n anstecken können und geheilt werden.

Die Inkubationszeit (Übergang von I zu M) wird als m bezeichnet.

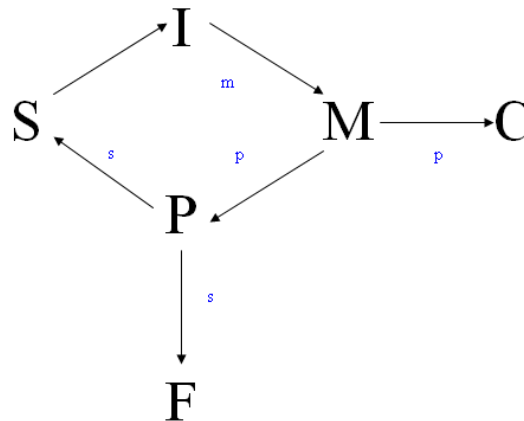
nun gilt: $G = M_n + S_n + I_n$

$$I_{n+1} = I_n + \eta * \frac{S_n}{S_n + I_n} * I_n - \eta * \frac{S_{n-m}}{S_{n-m} + I_{n-m}} * I_{n-m}$$

$$S_{n+1} = S_n - \eta * \frac{S_n}{S_n + I_n} * I_n$$

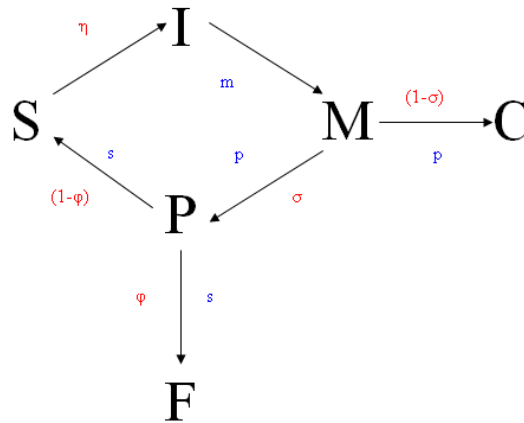
$$M_{n+1} = M_n + \eta * \frac{S_{n-m}}{S_{n-m} + I_{n-m}} * I_{n-m}$$

Weiters werden dem Modell noch folgende Klassen beigefügt: C (die Toten, von lat. *capulus* – Friedhof), P (die zwischenzeitlich Immunen, von lat. *prohibere* – abhalten) und F (die Immunen, von lat. *felix* – glücklich)



Ein Individuum kommt nach der Quarantänezeit p von der M -Klasse mit der Wahrscheinlichkeit σ (Sigma, von lat. *spes* – Hoffnung) in die P -Klasse und mit der Wahrscheinlichkeit $(1 - \sigma)$ zu C .

Ein P_n kommt nach der Immunitätszeit s mit der Wahrscheinlichkeit φ (Phi, von lat. *felix* – glücklich) in die Klasse F , in der es für immer immun bleibt. Es gehört zwar noch zur Gesamtpopulation und hat mit anderen Mitgliedern der Gesamtgruppe Kontakt, fällt aber dank seiner Immunität aus dem Krankheitszyklus. Mit der Wahrscheinlichkeit $(1 - \varphi)$ kehrt das Individuum wieder in die Klasse S zurück.



Zur Vervollständigung werden noch eine Zuwanderungs- beziehungsweise Geburtenrate α (Alpha: erster Buchstabe im griech. Alphabet – „der Anfang“) und ein natürlicher Sterbefaktor ω (Omega: letzter Buchstabe – „das Ende“) hinzugefügt.

Um eine Fehlerquelle beim Programmieren ausschließen zu können, wird der externe Überträger X eingeführt. X infiziert mit der Wahrscheinlichkeit η ein ansteckbares Individuum.

Formeln des Modells

Klassen		Parameter	Zeitspannen
X	Auslöser der Krankheit		n Anzahl der vergangenen Zeitpunkte
S	Ansteckbare	η Aggressionsfaktor	
I	Ansteckende	σ Hoffnungsfaktor	m I→M –Inkubationszeit
M	Morituri (in Quarantäne)	φ Glücksfaktor	p M→P bzw. M→C - Quarantänezeit
P	Zeitweise Immune	α Geburtenrate	s P→S bzw. P→F –vorübergehende Immunitätszeit
F	Immune	ω Sterberate	
C	Tote		

$$\bar{S}_n = \bar{F}_n = \frac{S_{n-m-p-s}}{S_{n-m-p-s} + I_{n-m-p-s} + P_{n-m-p-s} + F_{n-m-p-s}} * I_{n-m-p-s}$$

$$\bar{I}_n = \frac{S_n}{S_n + I_n + P_n + F_n} * I_n$$

$$\bar{M}_n = \frac{S_{n-m}}{S_{n-m} + I_{n-m} + P_{n-m} + F_{n-m}} * I_{n-m}$$

$$\bar{P}_n = \bar{C}_n = \frac{S_{n-m-p}}{S_{n-m-p} + I_{n-m-p} + P_{n-m-p} + F_{n-m-p}} * I_{n-m-p}$$

Erklärung der Formeln:

Die Krankheit wird durch den externen Überträger X ausgelöst, der zur Klasse I_1 addiert wird. Die Anzahl derer, die die Klasse wechseln, wurde als nicht linearer Ausdruck $\bar{I}_n$ modelliert, welcher die Kontakte von Ansteckenden und Ansteckbaren ausdrückt.

Die ansteckbaren Individuen (S_{n+1}) zur Zeit $n+1$ setzen sich aus den Ansteckbaren der letzten Messung (S_n), den Neugeborenen ($\alpha * (S_n + I_n + M_n + P_n + F_n)$) und denen, die aus der Klasse der zeitweise Immunen dazu stoßen ($(1-\varphi) * \sigma * \eta * \bar{S}$) zusammen, werden aber um die Anzahl derer verringert, die in die Klasse I wechseln ($\bar{I}_n$) oder eines natürlichen Todes sterben, ($\omega_s * S_n$).

$$S_{n+1} = S_n + (1-\varphi) * \sigma * \eta * \bar{S}_n - \eta * \bar{I}_n - \omega_s * S_n + \alpha * (S_n + I_n + M_n + P_n + F_n)$$

Die infizierten Individuen (I_{n+1}) zur Zeit $n+1$ setzen sich aus den Infizierten der letzten Messung (I_n) und denen zusammen, die aus der Klasse S dazu stoßen ($\eta * \bar{I}_n$), werden aber um die Anzahl derer verringert, die in die Klasse M wechseln ($\eta * \bar{M}_n$) oder eines natürlichen Todes sterben, ($\omega_I * I_n$).

$$I_{n+1} = I_n + \eta * \bar{I}_n - \eta * \bar{M}_n - \omega_I * I_n$$

Die Individuen in Quarantäne (M_{n+1}) zur Zeit $n+1$ setzen sich aus den Isolierten der letzten Messung (M_n) und den Infizierten, die vor m Messungen zu I gestoßen sind ($\eta * \bar{M}_n$), zusammen, werden aber um die Anzahl derer verringert, die in die Klasse P ($\sigma * \eta * \bar{P}_n$) oder durch die Krankheit in die Klasse C ($(1-\sigma) * \eta * \bar{C}_n$) wechseln oder eines natürlichen Todes sterben, ($\omega_M * M_n$).

$$M_{n+1} = M_n + \eta * \bar{M}_n - \eta * \sigma * \bar{P}_n - (1-\sigma) * \eta * \bar{C}_n - \omega_M * M_n$$

Die zeitweise Immunen (P_{n+1}) zur Zeit $n+1$ setzen sich aus den zeitweise Immunen der letzten Messung (P_n) und denen zusammen, die vor $m+p$ Messungen infiziert worden sind ($\sigma * \eta * \bar{P}_n$), verringert um die Anzahl, derer, die in die Klassen S ($(1-\varphi) * \sigma * \eta * \bar{S}_n$) oder F ($\varphi * \sigma * \eta * \bar{F}_n$) wechseln, oder eines natürlichen Todes sterben, ($\omega_p * P_n$).

$$P_{n+1} = P_n + \sigma * \eta * \bar{P}_n - (1-\varphi) * \sigma * \eta * \bar{S}_n - \varphi * \sigma * \eta * \bar{F}_n - \omega_p * P_n$$

Die ewig Immunen (F_{n+1}) zur Zeit $n+1$ setzen sich aus den ewig Immunen (F_n) der letzten Messung und denen, die vor $m+p+s$ Messungen infiziert worden sind ($\varphi * \sigma * \eta * \bar{F}_n$), zusammen, werden aber um die Anzahl derer verringert, die eines natürlichen Todes sterben, ($\omega_F * F_n$).

$$F_{n+1} = F_n + \varphi * \sigma * \eta * \bar{F}_n - \omega_F * F_n$$

Die Anzahl der Toten (C_{n+1}) zur Zeit $n+1$ entspricht der Anzahl der sich schon in der letzten Messung in C Befindenden (C_n), addiert mit der Menge aller krankheitsbedingten Tode ($(1-\sigma) * \eta * \bar{C}_n$) und natürlichen Todesfälle ($\omega_s * S_n + \omega_I * I_n + \omega_p * P_n + \omega_M * M_n + \omega_F * F_n$).

$$C_{n+1} = C_n + (1-\sigma) * \eta * \bar{C}_n + \omega_s * S_n + \omega_I * I_n + \omega_p * P_n + \omega_M * M_n + \omega_F * F_n$$

Beispiele

1. Influenza

Die "echte Grippe" kann unbehandelt eine tödliche Virusinfektion sein. Glücklicherweise gibt es einen Impfstoff gegen diese Infektion.

Das Virus wird entweder übertragen

- per **Tröpfcheninfektion** d.h. direktes Einatmen von Expirationströpfchen (Ausatmungströpfchen) infizierter Personen.
- über das **Trinkwasser** (Viren werden bei Kälte über mehrere Wochen konserviert)
- durch **Kotpartikel** erkrankter Wirte
- **mechanisch** auf Haaren, Haut und Gefieder

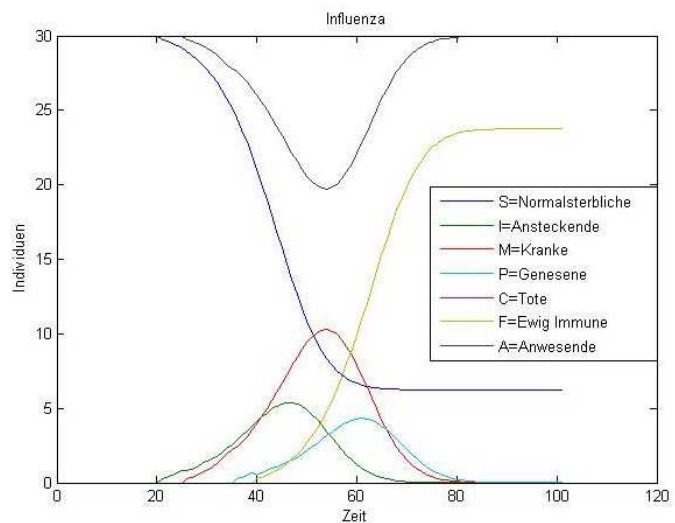
Es gibt unterschiedliche Schätzungen, nach welcher Zeit ein infiziertes Individuum seinerseits in der Lage ist, das Virus auf andere Individuen zu übertragen.

Es dauert 2-4 Tage bis ein gerade angesteckter Mensch weitere infizieren kann.

Influenza in einer Schulklass

Bei diesem Modell wird ein niedriger Aggressivitätsfaktor gewählt und somit verbreitet sich die Infektion nur langsam. Alle Schulkinder, bei denen die Krankheit ausgebrochen ist, werden behandelt (kommen nach M) und sterben deshalb nicht ($\sigma=1$). In weiterer Folge werden alle immun ($\varphi=1$), d.h., dass es in diesem geschlossenen System keine Neuerkrankungen gibt.

Da uns keine genauen Daten zur Verfügung standen, sind die Ergebnisse unserer Simulationen als qualitativ zu interpretieren.



2. Windpocken

Windpocken ist eine hoch ansteckende Infektionskrankheit, von der vor allem Kinder im Vorschulalter betroffen sind. Bei 90% der Infizierten kommt es zu einer lebenslangen Immunität.

Die Inkubationszeit bei Windpocken liegt zwischen zehn und 21 Tagen. Die Krankheit bleibt ca. 7-10 Tage nach Bildung der ersten Bläschen auf der Haut ansteckend. In dieser Zeit sollte die erkrankte Person nicht in Kontakt mit anderen kommen, vor allem nicht mit älteren Menschen oder Schwangeren, da bei diesen Personen die Krankheit schwerwiegendere Folgen haben kann.

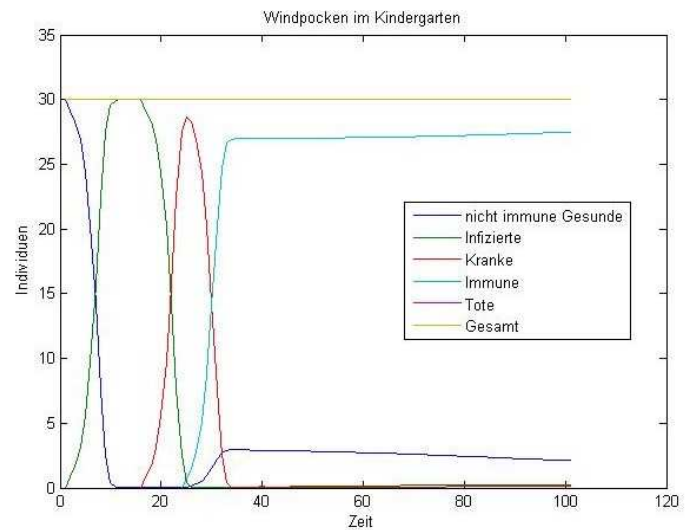


a) Windpocken im Kindergarten

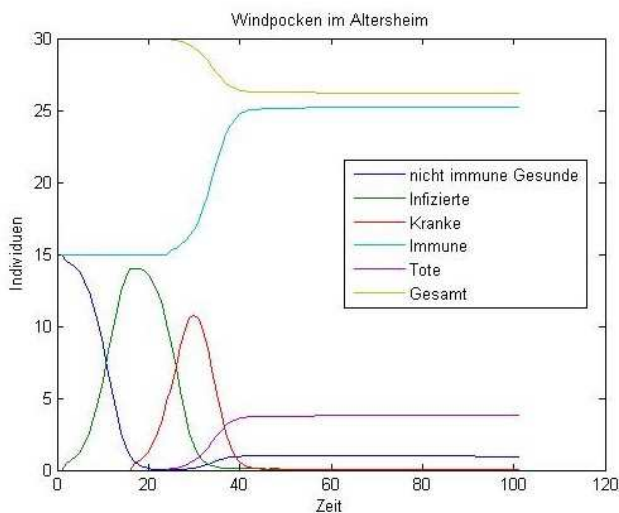
In einem Kindergarten befinden sich 30 gesunde, nicht gegen Windpocken immune Kinder (S). Es kommt zur Infektion eines einzelnen Kindes, das die Epidemie ins Rollen bringt. Da noch keines der anderen Kinder bereits zuvor an Windpocken erkrankt war, kommt es im Laufe der Zeit bei allen zur Infektion.

Im Modell wird auf Grund von statistische Daten von einer ständigen Immunität bei 90% der Erkrankungen ausgegangen ($\phi=0,9$). Die Sterberate liegt bei 0% ($\sigma=1$). Weiters werden die Parameter für die Inkubationszeit ($m=15$) und die Krankheitsdauer ($p=8$) angeglichen. Da Windpocken eine sehr ansteckende Krankheit ist wurde der Aggressionsfaktor extrem hoch angelegt ($\eta=0,9$). Geburten- und natürliche Sterberate werden ganz ignoriert.

Ergebnis: Nach einem einmaligen starken Ausbruch der Epidemie, verschwindet die Krankheit vollkommen.



b) Windpocken im Altersheim



In einem Altersheim befinden sich 30 Personen, von denen 15 bereits gegen Windpocken immun sind. Es beginnt wieder mit einer Infektion, die eine Epidemie auslöst. Da Windpocken bei älteren – vielleicht kranken – Personen eine größere Gefahr darstellt als bei Kindern, wird σ mit 0,75 festgelegt.

Ein weiterer Unterschied liegt noch bei der natürlichen Sterberate (ω) vor, die hier nicht ignoriert wird, sondern zwischen 0,02 und 0,05 schwankt.

Die Parameter m , p , s , ϕ und η werden im Vergleich zum Kindergartenmodell nicht verändert.

Weitere Beispiele

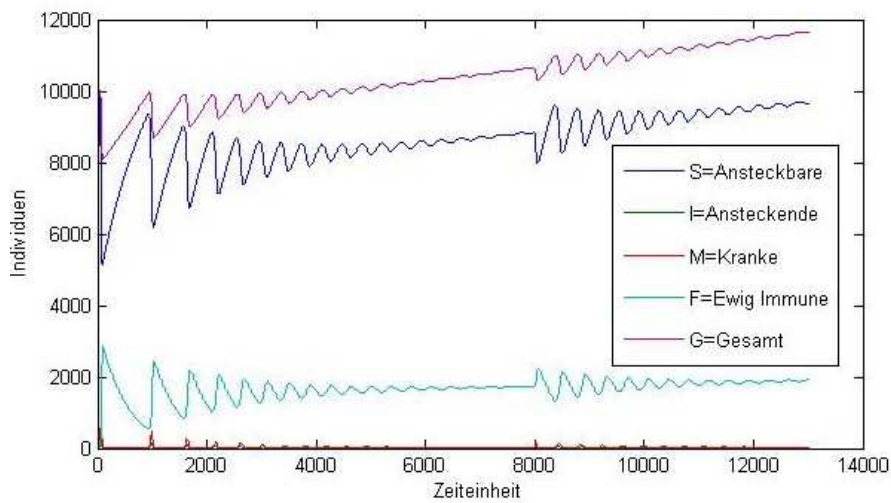


Abbildung 1: Zwei Epidemien hintereinander

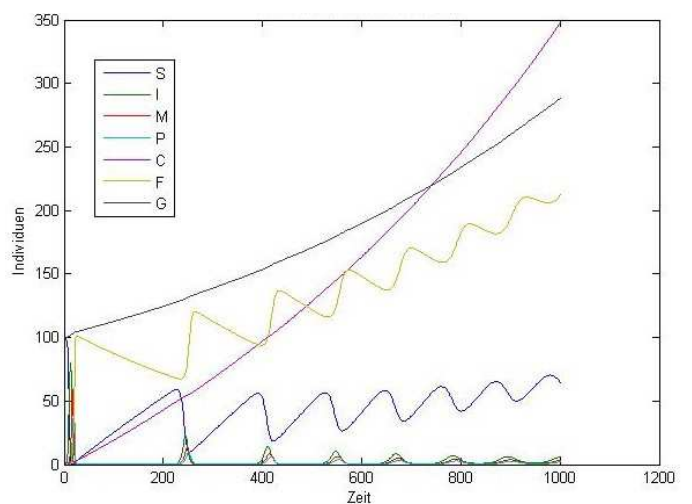


Abbildung 2: Epidemien treten immer wieder abgeschwächt auf

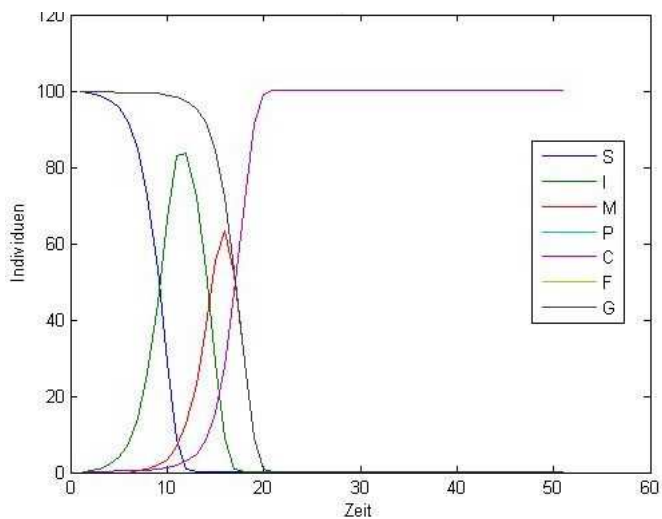
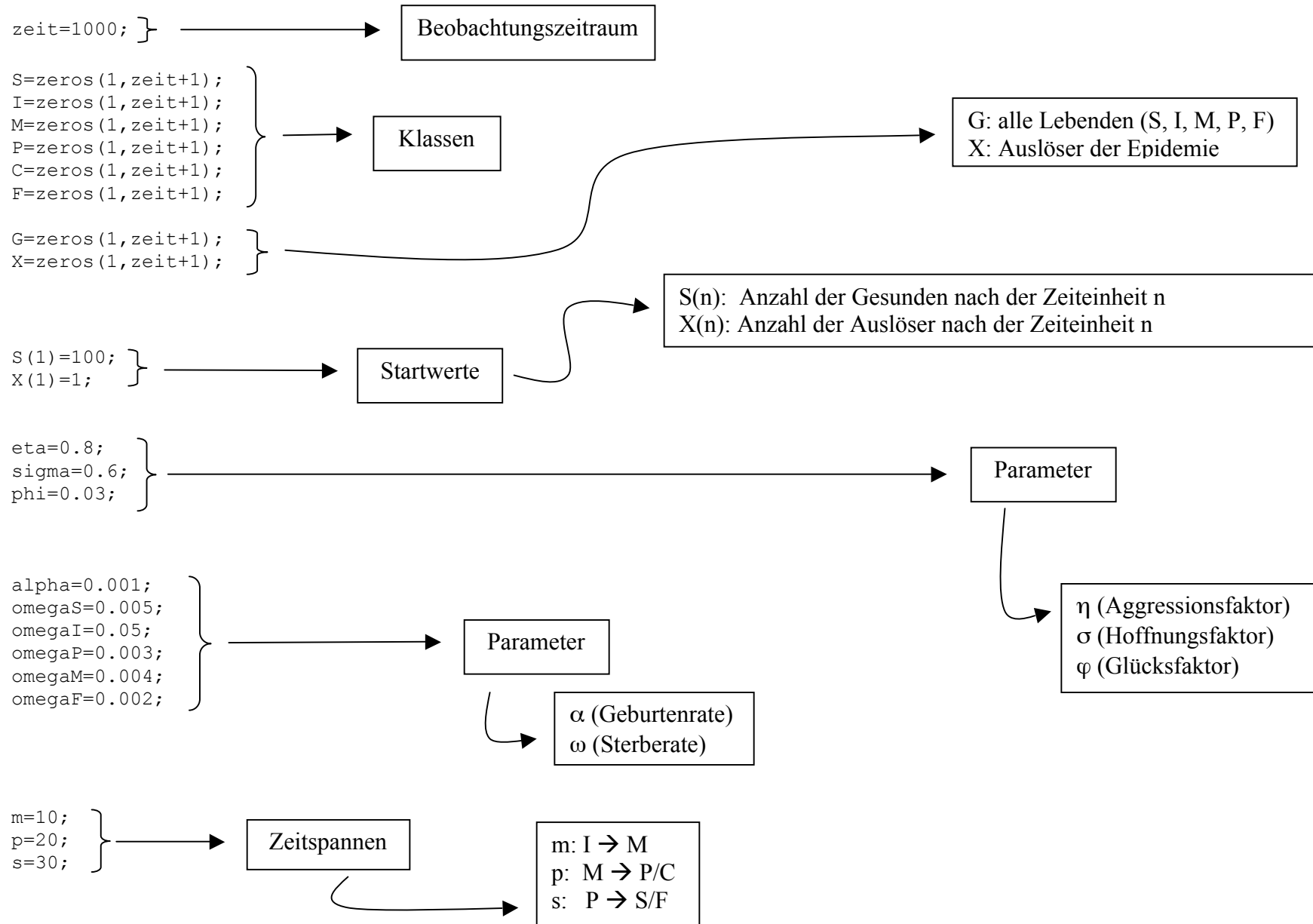


Abbildung 3: Ausrottung der Individuen

Programm





```
for n=(1:zeit)
```

```
Iw=eta*S(n)*(I(n)+X(n))/(S(n)+I(n)+P(n)+F(n));
```

```
}
```

Auslöser der Epidemie
kommt ins Spiel

```
Mw=0;
```

```
Pw=0;
```

```
Cw=0;
```

```
Sw=0;
```

```
Fw=0;
```

```
}
```

Als I_w , M_w , P_w , C_w , S_w und F_w werden jene Individuen
bezeichnet, die in die jeweilige Gruppe wechseln.

z.B. besteht die Klasse I nach einer Zeiteinheit aus folgenden
Elementen:

Alle außer I_w (wegen X, dem
ersten Infizierten) haben als
Startwert 0.

```
if I(n+1)>0
```

```
    I(n+1)=I(n+1)-omegaI*I(n);
```

```
    C(n+1)=C(n+1)+omegaI*I(n);
```

```
end
```

```
if S(n+1)>0
```

```
    S(n+1)=S(n+1)-omegaS*S(n);
```

```
    C(n+1)=C(n+1)+omegaS*S(n);
```

```
end
```

```
if P(n+1)>0
```

```
    P(n+1)=P(n+1)-omegaP*P(n);
```

```
    C(n+1)=C(n+1)+omegaP*P(n);
```

```
end
```

```
if M(n+1)>0
```

```
    M(n+1)=M(n+1)-omegaM*M(n);
```

```
    C(n+1)=C(n+1)+omegaM*M(n);
```

```
end
```

```
if F(n+1)>0
```

```
    F(n+1)=F(n+1)-omegaF*F(n);
```

```
    C(n+1)=C(n+1)+omegaF*F(n);
```

```
end
```

$$I_{n+1} = I_n + I_w - M_w - \text{Sterbefaktor}$$

I_n : Individuen, die bereits zuvor in I waren
 I_w : Individuen, die nach I wechseln
 M_w : Individuen, die aus I nach M wechseln

Sterberaten (ω) werden erst anschließend
berechnet, um die Stabilität des Programms zu
garantieren und zwar nur dann, wenn sich
Individuen in einer Klasse befinden.

```

if n>m
    Mw=eta*S(n-m)*(I(n-m)+X(n-m))/(S(n-m)+I(n-m)+P(n-m)+F(n-m));
end

if n>m+p
    Pw=sigma*eta*S(n-m-p)*(I(n-m-p)+X(n-m-p))/(S(n-m-p)+I(n-m-p)+P(n-m-p)+F(n-m-p));
    Cw=(1-sigma)*eta*S(n-m-p)*(I(n-m-p)+X(n-m-p))/(S(n-m-p)+I(n-m-p)+P(n-m-p)+F(n-m-p));
end

if n>m+p+s
    Sw=(1-phi)*sigma*eta*S(n-m-p-s)*(I(n-m-p-s)+X(n-m-p-s))/(S(n-m-p-s)+I(n-m-p-s)+P(n-m-p-s)+F(n-m-p-s));
    Fw=phi*sigma*eta*S(n-m-p-s)*(I(n-m-p-s)+X(n-m-p-s))/(S(n-m-p-s)+I(n-m-p-s)+P(n-m-p-s)+F(n-m-p-s));
end

I(n+1)=I(n)+Iw-Mw;
M(n+1)=M(n)+Mw-Pw-Cw;
P(n+1)=P(n)+Pw-Sw-Fw;
C(n+1)=C(n)+Cw;
S(n+1)=S(n)+Sw-Iw+alpha*(S(n)+I(n)+M(n)+P(n)+F(n));
F(n+1)=F(n)+Fw;
G=I+M+P+S+F;
end

```

Definition der Klassen und des
Krankheitskreislaufes

Geburtenrate

```

plot([1:zeit+1],S,[1:zeit+1],I,[1:zeit+1],M,[1:zeit+1],P,[1:zeit+1],C,[1:zeit+1],F,[1:zeit+1],G)
legend('S','I','M','P','C','F','G')

```

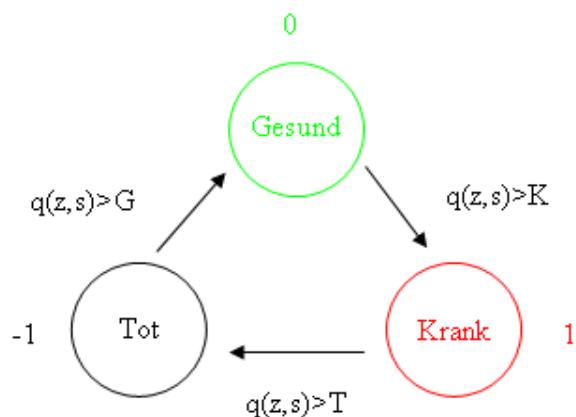
Modell 2: „Der kranke Wald“

Dieses Modell ist im Gegensatz zum ersten eines mit unbeweglichen Individuen, das mit Hilfe eines Zellularautomaten simuliert wird.

Es wird die flächenmäßige Ausbreitung einer Baumkrankheit in einem Wald (abgeschlossenes System) dargestellt.

In jeder Zelle kann entweder ein gesunder (0), kranker (1) oder toter (-1) Baum stehen.

Zu Beginn sind fast alle Bäume gesund, und nur einige wenige infiziert, die wiederum unter bestimmten Bedingungen (genügend kranke Nachbarn) gesunde Bäume infizieren.



Die kranken können zu toten Bäumen werden, wenn q (eine zufällige Zahl zwischen 1 und 0) größer als die festgelegte Wahrscheinlichkeit (T) ist, d.h. nicht alle erkrankten Bäume sterben zwangsläufig nach einer Zeiteinheit (n).

Nachdem ein Baum gestorben ist, kann – wenn $q > G$ – ein neuer, gesunder Baum nachwachsen.

Die Möglichkeit, dass ein gesunder Baum erkrankt, steigt mit der Anzahl der kranken Nachbarbäume ($q * Anzahl > K$).

Unser Simulationsprogramm berechnet die Zustände 0, 1, -1 als Einträge einer

großen rechteckigen Matrix zu jedem diskreten Zeitpunkt. Das Update geschieht gemäß den angegebenen Regeln. Die dynamische Entwicklung der Epidemie wird in einer bewegten, dreifarbigen Grafik dargestellt, nämlich als grüne, gesunde Zellen, rote, kranke Zellen und schwarze, tote Zellen.

Durch Verändern der Parameter und Berücksichtigung des Zufallsgenerators können verschiedenste Szenarien simuliert werden, z. B. das Aussterben der Krankheit, oder diverse Ausbreitungsarten der Krankheit.

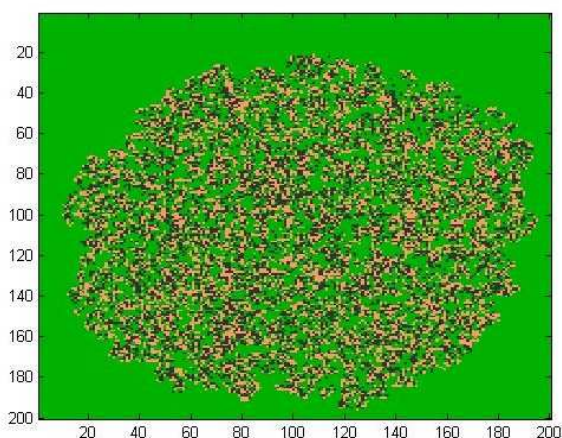


Abbildung 1

Abbildung 1 und 2

Harmlose Krankheit, die in der Simulation nicht ausstirbt; Bäume werden nur geschädigt.

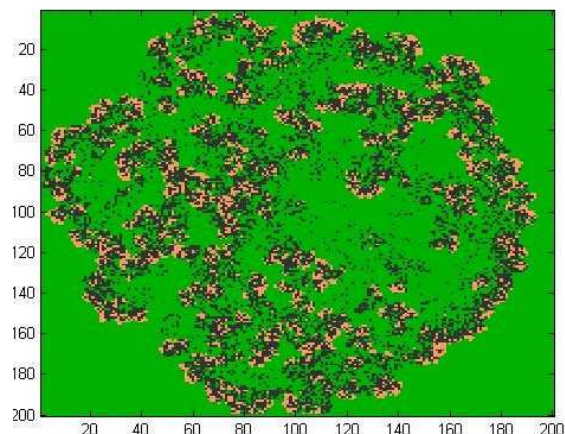


Abbildung 1

Abbildung 3
Waldbrand

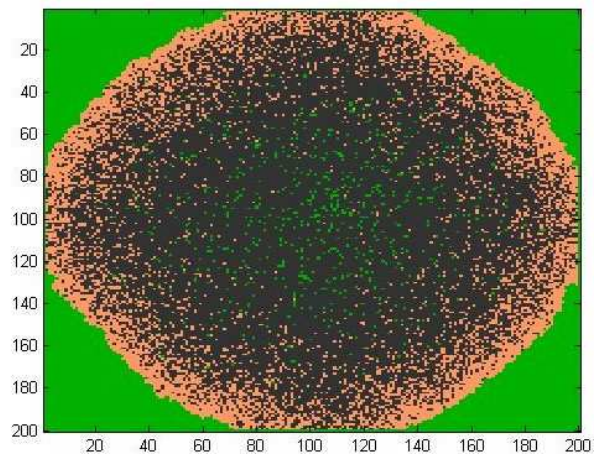


Abbildung 3

Abbildung 4
Insektenplage (z.B. Borkenkäfer)

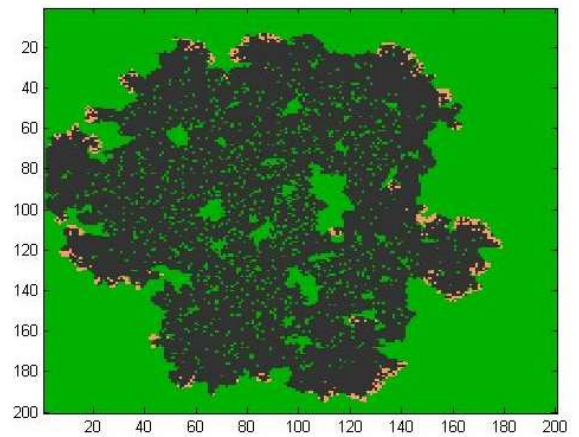


Abbildung 4

Programmbeschreibung Waldmodell

```
colormap([0.2 0.2 0.2; 0 0.7 0; 1 0.6 0.4]);
```

```
sg=200;  
zg=200;
```

Länge und Breite des Waldes.

```
q=zeros(zg, sg)  
Wald1=zeros(zg, sg);  
Wald=zeros(zg, sg);
```

Es werden drei Matrizen (Wald, Wald1 und q) mit der angegebenen Größe gebildet.

```
Wald(round(sg/2)-1, round(zg/2))=1;  
Wald(round(sg/2), round(zg/2))=1;  
Wald(round(sg/2)+1, round(zg/2))=1;
```

Drei Zellen in der Mitte der Matrix werden als „Infizierte Bäume“ festgelegt.

```
K=1;  
T=0.96;  
G=0.999;
```

Parameter: K - Infektionswahrscheinlichkeit
T - Sterbewahrscheinlichkeit
G - Nachwuchswahrscheinlichkeit

```
for n=[1:200]  
    q=rand(zg, sg);  
    Wald1=Wald;
```

Hier werden 200 Durchgänge definiert. In die Matrix q werden Zufallswerte geschrieben, Wald1 erhält alle Werte von Wald.

```
for z=[2:zg-1]  
    for s=[2:sg-1]
```

Alle Zellen der Matrix, außer die am Rand, werden der Reihe nach aufgerufen.

```
        anz=max(0, Wald(z-1, s))+max(0, Wald(z+1, s))...  
        +max(0, Wald(z, s-1))+ max(0, Wald(z, s+1))...  
        +max(0, Wald(z-1, s-1))+max(0, Wald(z-1, s+1))...  
        +max(0, Wald(z+1, s-1))+max(0, Wald(z+1, s+1));
```

Bei allen Bäumen werden deren benachbarte infizierte Bäume gezählt.

```
        if Wald(z, s)==0 & anz*q(z, s)>K  
            Wald1(z, s)=1;  
        end
```

Gesunde Bäume werden mit Wahrscheinlichkeit infiziert, wenn infizierte Nachbarn vorhanden sind.

```
        if Wald(z, s)==1 & q(z, s)>T  
            Wald1(z, s)=-1;  
        end
```

Kranke Bäume sterben mit Wahrscheinlichkeit ab.

```
        if Wald(z, s)==-1 & q(z, s)>G  
            Wald1(z, s)=0;  
        end
```

Tote Bäume wachsen mit Wahrscheinlichkeit nach.

```
    end  
end
```

```
Wald=Wald1;  
image(Wald+2);  
drawnow  
end
```

Alle Informationen über den nachfolgenden Zustand wurden in der Matrix „Wald1“ gespeichert. Nun werden diese Informationen auf die Matrix „Wald“ übertragen. Diese wird danach graphisch dargestellt.

Projekt: Bildverarbeitung, Interpolation von Bildern

13. Januar 2006

Inhaltsverzeichnis

1 Teilnehmer	3
2 Leitung	3
3 Problem	3
3.1 Problemstellung	3
3.2 Überlegungen	3
3.3 Verwendete Programme	3
4 Problemlösungsversuche	4
5 Problemlösung	4
5.1 Vorarbeit	4
5.1.1 JGP2Matrix	4
5.1.2 Berechnung des Schwerpunktes	4
5.1.3 Segmentierung	4
5.1.4 Registrierung	5
5.2 Problemlösung	5
5.2.1 Methode 1: Uhrzeiger-Algorithmus	5
5.2.2 Methode 2: Thin-Plate Spline Algorithmus	6
6 Epilog	6
6.1 Fakten	7
7 Quellcode	7

1 Teilnehmer

Michael Hsieh, Daniel Kraft, Sara Kropf, Christoph Planitzer, Stefan Reisinger, Christoph Triebel

2 Leitung

Mag. Dr. Stephen Keeling



3 Problem

3.1 Problemstellung

Zeitreihen von medizinischen Bildern werden oft aufgenommen, um physiologische Prozesse dynamisch zu untersuchen. Damit solche Prozesse immer detaillierter untersucht werden können, wird der Bedarf für hohe zeitliche Auflösung immer größer, d. h. die Anzahl von aufgenommenen Bildern pro Zeiteinheit muss immer größer werden. Andererseits verlangt die Aufnahme eines einzigen Bildes eine gewisse Zeitdauer. Wenn diese Zeitdauer relativ groß ist, wird die dynamische Untersuchung verhindert. Es stellt sich dann die Frage, ob man zwischen Bildern der aufgenommenen Zeitreihe interpolieren kann, um die dynamische Untersuchung zu erleichtern. Das Ziel dieses Projekts ist es, ein Verfahren zu entwickeln, mit dem fehlende Bilder zwischen zwei gegebenen Bildern interpoliert werden können.

3.2 Überlegungen

Schon am Anfang stellten sich große Hürden: Das erste Problem war, Objekte vom Hintergrund der Fotos zu segmentieren¹ und einander zuzuordnen. Ein großer Meilenstein war die Überlegung, den Schwerpunkt eines Objektes als Ursprung eines kartesischen Koordinatensystems zu nutzen. Ein weiteres Problem war die Antwort auf die Frage, ob der Zeitunterschied zwischen den Bildern linear ist, sprich der Zeitabstand zwischen Anfangs- und eingefügtem Bild gleich groß ist wie der Zeitabstand zwischen eingefügtem Bild und Endbild. Schwierig war es für uns auch, das von Keeling gezeigte Programm, welches die Annäherung einer Geraden an eine andere zeigte, für unsere Bilder zu verwenden. Ein anderes Problem war die Bewältigung des Übergangs von einfachen linearen Funktionen, sprich 1D, auf komplexe Bilder (2D).

3.3 Verwendete Programme

- Matlab
- C++
- Freeimage Library
- gcc
- gtk+
- gtkmm

¹unterscheiden

4 Problemlösungsversuche

Eingesperret im letzten Kellerloch, düster, die Fenster mit Jacken abgedunkelt, die Seiten des Flipcharts noch völlig unbeschrieben, nur mit wenig Mineralwasser, welches wir Zwecks Kühlung mit dem Stromkabel unsers verwaahrlosten Overheadprojektors am Fenstergitter herabhängen ließen, für das Aufrechterhalten unseres Kreislaufsystems versorgt, stolz auf unser Raumnetzwerk und ausgestattet mit 5 Laptops und einem iBook, gingen wir nach diesen reichlich durchdachten Überlegungen an die eigentliche Arbeit. Wir befassten uns mit dem Problem, wie ein Körper sich bewegen und verändern muss, um vom Anfangs- zum Endbild zu gelangen. Wir versuchten, ein links oben liegendes Quadrat nach rechts unten zu transformieren. Hierfür verwendeten wir parallele Vektoren, die das Quadrat an die gewünschte Stelle brachten. Wir stießen auf das Problem, dass bei komplizierten Körpern die Möglichkeit bei einer Drehung besteht, dass sich alle diese Vektoren im Extremfall in einem Punkt schneiden und so der Körper zu einem Punkt deformiert wird.

Danach suchten wir einen Weg, die Position eines Randpunktes eines Körpers im eingefügten Bild genau zwischen Ausgangs- und Endposition zu platzieren. Doch leider beachteten wir nicht, dass der Computer nicht erkennen kann, wo sich ein beliebiger Punkt des Ausgangsbild in den anderen Bildern befindet.

Als wir auf die Idee kamen, dass der Computer bestimmte Farbunterschiede erkennen und mit dem Endbild vergleichen könnte, haben wir nicht in Erwägung gebracht, dass sich Körper auch drehen können und somit die dunklen Flächen am oberen Rand des ersten Bildes nicht den dunklen Flächen am oberen Rand des zweiten Bildes entsprechen müssen.

$$F(\bar{p}) = \int (I_0(u(\bar{p}) - I_1(x))^2)$$

Mit dieser Formel versuchten wir die Abweichung zweier Bilder zu berechnen.

So rätselten wir einige Zeit weiter...

5 Problemlösung

5.1 Vorarbeit

5.1.1 JGP2Matrix

Um Bilder bearbeiten zu können, müssen sie in Form einer Matrix dargestellt werden. Wir schrieben ein Programm in C++, welches uns dies ermöglicht.

5.1.2 Berechnung des Schwerpunktes

Der Schwerpunkt im Zwischenbild wird durch das gewichtete Mittel der zwei ursprünglichen Schwerpunktkoordinaten berechnet. Analog dazu berechnen wir den Radius eines Punktes des Zwischenbildes im Polarkoordinatensystem. Auf diese Weise ermitteln wir einander zugeordnete Punkte der Bilder.

$$\bar{x} = \frac{\int (I \cdot x \cdot dx) dy \bar{y}}{\int (I \cdot y \cdot dx) dy}$$

5.1.3 Segmentierung

Die Segmentierung dient zur Identifikation von einander entsprechenden Objekten. Hierfür machten wir Gebrauch von der Level-Set-Methode.

Level Sets Die Level-Set-Methode oder Niveaumengenmethode ist ein mathematisches Verfahren um geometrische Objekte und deren Bewegung numerisch zu berechnen. Der Vorteil der Level-Set-Methode liegt darin, dass man Kurven und Oberflächen auf einem räumlich festen (Eulerschen) Koordinatensystem berechnen kann, ohne Parametrisierungen dieser Objekte verwenden zu müssen. Insbesondere muss bei der Level-Set-Methode die Topologie (zum Beispiel die Anzahl der zusammenhängenden Gebiete) nicht bekannt sein, und sie kann sich während der Berechnung ändern. Dies erlaubt die einfache Verfolgung der Ränder beweglicher Objekte, beispielsweise eines Airbags oder eines Tropfens Öl, der in Wasser schwimmt. Bei der Level-Set-Methode wird im n-dimensionalen Raum ein (n-1)-dimensionaler Rand Γ (etwa eine Kurve für $n=2$) als Nullstellenmenge (level-set) einer n-dimensionalen Hilfsfunktion ϕ beschrieben:

$$\Gamma(\vec{x}) = \{\vec{x} | \phi(\vec{x}) = 0\}$$

Flatfill-Algorithmus Dabei handelt es sich um ein anderes, sehr einfaches Segmentierungsverfahren, dass zusammenhängende Flächen von Pixeln mit ähnlichen Farbwerten findet. Von einem Anfangspunkt ausgehend wird das Verfahren rekursiv auf die 4 Nachbarpixel angewendet, bis der gesamte Inhalt des Objekts identifiziert wurde.

5.1.4 Registrierung

Die Registrierung erkennt einander zugeordnete Objekte in den beiden Bildern und berechnet die Koordinatentransformationen.

5.2 Problemlösung

5.2.1 Methode 1: Uhrzeiger-Algorithmus

Der Umriss der Bilder ist als Funktion in Polarkoordinaten² gegeben. Der Ursprung des Koordinatensystems liegt im Schwerpunkt des Objektes. Durch das PCA-Verfahren (s.u.) ist der Winkel θ gegeben. Die folgende Funktion liefert das interpolierte Bild im Polarkoordinatensystem.

$$r(\phi + t \cdot \theta) = (1 - t) \cdot r_1(\phi) + t \cdot r_2(\phi + \theta)$$

Der Radius in Abhängigkeit des Winkels ist gegeben als das gewichtete arithmetische Mittel der entsprechenden Radien der Ausgangsbilder.

Subfunktionen

winkel() berechnet den Winkel mit Hilfe von Schwerpunkt(), Covariance() und trans().

procObj() Hauptfunktion, führt Abtasten für ein Objekt durch

radius() berechnet den Abstand vom Schwerpunkt bis zum Rand für einen bestimmten Winkel.

radialSchnitt() berechnet ein eindimensionales Bild entlang einer Schnittlinie

radiusInterpol() liefert das 1D-Bild für das interpolierte Bild.

schreibeSchnitt() fügt das errechnete 1D-Bild ins Zielbild ein.

PCA-Verfahren Mit Hilfe des PCA-Verfahrens können wir die Richtung der Koordinatenachsen so bestimmen, dass eine der Achsen eine maximale Länge innerhalb der Figur aufweist. Damit kann eine Drehung der Figur dargestellt werden.

Wikipedia: Principal Component Analysis: In diesem Verfahren versucht man, aus Daten mit vielen Eigenschaften einige wenige latente Faktoren zu extrahieren, die für diese Eigenschaften bestimmend sind. Mathematisch wird eine Hauptachsentransformation durchgeführt: Man minimiert die Korrelation mehrdimensionaler Merkmale durch Überführung in einen Vektorraum mit neuer Basis. Die Hauptachsentransformation lässt sich durch eine Matrix angeben, die aus den Eigenvektoren der Kovarianzmatrix gebildet wird. Die Hauptkomponentenanalyse ist damit problemabhängig, weil für jeden Datensatz eine eigene Transformationsmatrix berechnet werden muss.

²Radius und Winkel

5.2.2 Methode 2: Thin-Plate Spline Algorithmus

Eine weitere Problemlösung stellt der Thin-Plate Spline Algorithmus dar. Ein selbstprogrammiertes Punktzuordnungsverfahren, das in Start- und Endbildern markante Punkte erkennt, übergibt diese zugeordneten Punkte an den Algorithmus, um eine Interpolation zwischen den Bildern durchzuführen. Außerdem programmierten wir ein GUI³ mit GTK+⁴ und gtkmm⁵ um die Punkterkennung manuell zu ermöglichen.

Subfunktionen

spliceLord() Lotst die Bildteile durch einen Registrierungsprozess, um anschließend eine Interpolation durchzuführen.

cronLord()

identify() hat die Aufgabe, Flächen einander zuzuordnen

getSignificantPoints() findet Punkte, nach denen interpoliert werden kann und nutzt die Funktion

getCrossingPoints() die die Kreuzungspunkte von Flächen berechnet.

getAccess() wendet das PCA-Verfahren auf die einzelnen Flächen an, um ihre Identität zu verifizieren. Wird von **identify()** aufgerufen.

morph() übernimmt schlussendlich die eigentliche Interpolation. Die Unterfunktionen **transform()**, **getParams()** und **getAccess()** vollziehen den von uns erarbeiteten Algorithmus.

6 Epilog

Erschöpft von einer Woche reiner mathematischer Probleme, mit denen wir unsere Gehirne stark beanspruchten, alle zur Verfügung gestellten Blöcken bis auf die letzte Seite mit Formeln und Skizzen bekritzelt, das Flipchart beschmiert bis aufs letzte Blatt, müde von zahlreichen durchgearbeiteten Nächten, blickten wir eingesperrt hinter dicken Kellermauern am letzten Abend sehnsüchtig nach Freiheit aus den zwei kleinen, vergitterten Luken der Sonne, die mit RGB(0xFF0000) hinter dem Horizont versank, entgegen.

³ Graphical User Interface

⁴ Gimp ToolKit

⁵ C++ Interface zu GTK

⁵Registrierung

6.1 Fakten

- Geleerte Mineralwasserflaschen: 30
- Netzwerktraffic (geschätzt): 3GB
- Beschriebene A4-Seiten: 100
- Beschriebene Flipchart-Seiten: 27
- Gelöste mathematische Rätsel: 10
- Vorbereitete Doktorarbeiten: 1
- Notendurchschnitt unserer Gruppe: 1.1
- Anzahl getippter Codezeichen: 57.361
- Wachgeküsste Apple: 1
- Gegessene Apple: 5
- Am häufigsten gesagte Wörter:
 1. Apple-Taste
 2. Freak
 3. Windows ist ...
 4. asozial
- Programmabstürze
 - Windoof: 21 (was war zu erwarten?)
 - Apple: 2 (Anwenderfehler/falscher Anwender)
 - Linux: 0 (was war zu erwarten?)
- GIMP-Installationen unter Windows: 8
- Endlosschleifen: 5
- Verhältnis benötigte Rechenzeit Matlab:C = 30:1
- Höhe der Kellertür: 3 Meter
- Vorhandene Netzwerksteckdosen: 6
- Funktionierende Netzwerksteckdosen: 1
- Laptops im Netz: 5
- Über Kabel gestolperte Personen: 14
- Eingefrorene Kabel: 3
- Raumtemperatur in der Früh: 30°C
- Durschnittl. Arbeitszeit pro Tag und Person: 13 Stunden

7 Quellcode

Der gesamte Quellcode ist frei unter der GPL-Lizenz verfügbar und kann unter folgender Adresse abgerufen werden: <http://bgweiz.at/reisi/interpol-quellcode/>