

## WOCHE

DER

## MODELLIERUNG

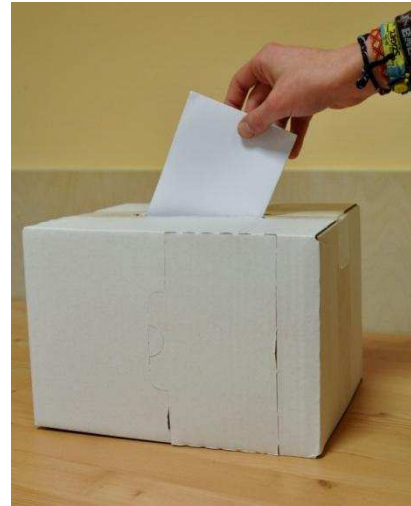
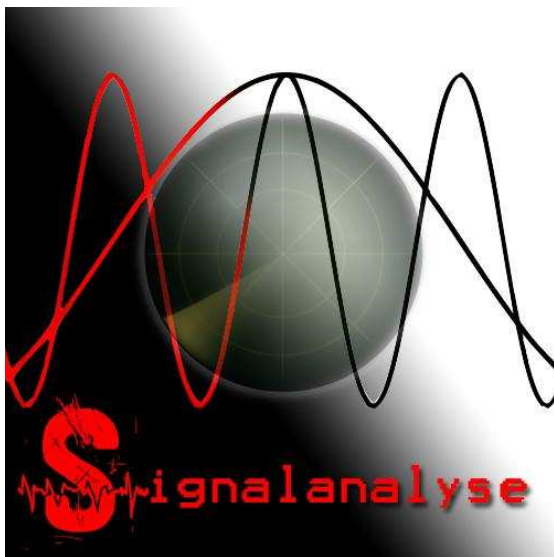
MIT

## MATHEMATIK

PageRank-Algorithmus

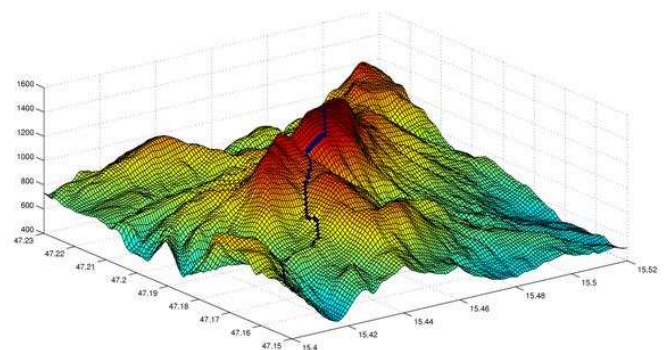
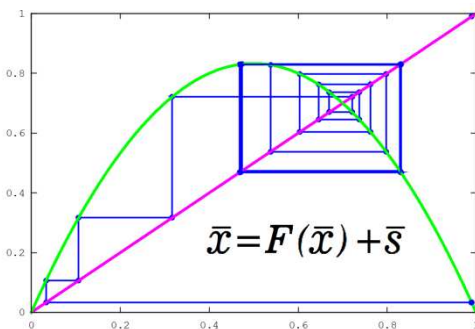
Google-Suche

Auf gut Glück!



### KURZBERICHT

6.2. – 12.2.2011



# WOCHE DER MODELLIERUNG MIT MATHEMATIK



**Pöllau bei Hartberg, 6.2. – 12.2.2011**

**Ausführliche Dokumentation der einzelnen Projekte:**

<http://math.uni-graz.at/modellwoche/2011/>



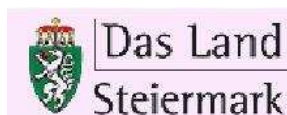
# Vorwort

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

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

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

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

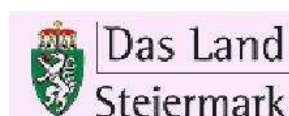


Besonderer Dank gebührt ferner Herrn Mag. Christoph Gruber, der die ganze Veranstaltung betreut und auch die Gestaltung dieses Berichtes übernommen hat, Frau Michaela Seiwald für die tatkräftige Hilfe bei der organisatorischen Vorbereitung, und Herrn Dr. Georg Propst für die Hilfe bei der Betreuung der Hard- und Software. Es wird auch heuer die Modellierungswoche selbst wissenschaftlich untersucht. Die fachdidaktische Begleitforschung wird von Herrn Mag. Christoph Gruber im Rahmen einer Dissertation an unserem Institut und im Auftrag des Bundesministeriums für Wissenschaft und Forschung durchgeführt. Wir danken den Schülerinnen und Schülern für ihre Geduld beim Ausfüllen zahlreicher Fragebögen und für ihre Bereitschaft, sich psychologischen Tests aller Art zu unterziehen.

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

Pöllau, am 12. Februar 2011

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



Projekt Signalanalyse

# Geschwindigkeits- und Entfernungsmessung mit Signalwellen

Dr. Peter Schöpf

TeilnehmerInnen: Nikolaus Leopold, Fabian Peter Hammerle, Dario Kaylani, Dino Mehic, Mario Messiha, Lissa Wilding

## PROBLEMSTELLUNG:

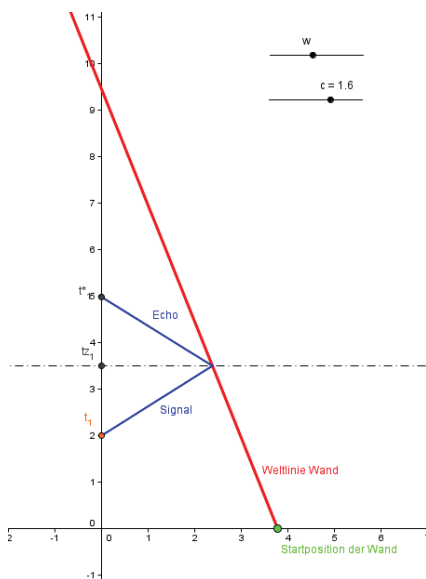
Signalausbreitung und -analyse spielen eine wichtige Rolle bei Radarmessungen. Dabei müssen wir zwei Arten der Messung unterscheiden: Sendung von Impulsen und deren Echos, oder dauerhafte Aussendung von Wellen konstanter Frequenz. Die Impulse haben alle die gleiche Zeitdauer (PW) und werden in gleichmäßigen Abständen (PRT) ausgesendet. Ein Messvorgang besteht in der Aussendung von ca. 100 solchen Impulsen und der Registrierung ihrer Echos.

Im Rahmen der Modellierungswoche setzten wir uns das Ziel die Welt der Physik mit der Mathematik zu verbinden. Um uns den anschaulichen Einstieg in die Probleme zu erleichtern, modellierten wir die auftretenden Bewegungsvorgänge zunächst mit GeoGebra. Anschließend leiteten wir allgemein gültige Formeln für einige Fragestellungen her. Insbesondere bestimmten wir die Gültigkeitsbereiche für Ort und Geschwindigkeit des bewegten Objektes, die für ein korrektes Funktionieren des Messgerätes notwendig sind.

## ERGEBNISSE:

### 1 Radarmessung

#### a) Umrechnungsfaktor für die Zeitintervalle



Einfache geometrische Überlegungen waren notwendig um die Eintreffzeit  $t^*$  des reflektierten Echosignals zu berechnen. Unter der Annahme, dass die Wandgeschwindigkeit  $w$  (d.h. die Geschwindigkeit eines bewegten Objekts) und die Signalgeschwindigkeit  $c$  gegeben sind, berechneten wir zunächst den Schnittpunkt des Signalstrahls mit der „Wand“ um daraufhin die Formel für die Eintreffzeit  $t^*$  zu bestimmen:

$$t^z = \frac{x(0) + c \cdot t}{c - w}$$

$c$  ... Lichtgeschwindigkeit  
 $w$  ... Wandgeschwindigkeit

Gehen wir von zwei hintereinander gesandten Signalen aus und deren Sendezeiten und Eintreffzeiten der Echos, so erhalten wir für die Geschwindigkeit  $w$  und den Ort  $x(0)$  folgendes:

- Wandgeschwindigkeit:  $w = \frac{\Delta t^* - \Delta t}{\Delta t^* + \Delta t} \cdot c$
- Startpunkt der Wand  $w$  zum Zeitpunkt  $t$ :  $x(0) = \frac{t_1^* \cdot t_2 - t_2^* \cdot t_1}{\Delta t^* + \Delta t} \cdot c$

- Verhältnis der Zeitdifferenzen:  $\Delta t^* = \frac{c+w}{c-w} \cdot \Delta t$

### b) Zulässiger Geschwindigkeitsbereich

Da für eine zuverlässige Messung die Echos zwischen den Signalimpulsen liegen müssen, stellten wir Ungleichungen auf, in denen sich der Rückkehrzeitpunkt der Echos befinden muss:

$$\begin{aligned} \text{1. Signal:} & \quad t_1 + PW \leq t_1^* \leq t_2 - EW \\ \text{n. Signal:} & \quad t_n + PW \leq t_n^* \leq t_{n+1} - EW \end{aligned}$$

Durch Einsetzen bereits bekannter Formeln ergab sich folgender, für die Geschwindigkeit zulässige, Bereich:

$$\frac{(2PW - PRT)}{PRT \cdot (2n - 3)} \cdot c \leq w \leq \frac{PRT - 2PW}{PRT \cdot (2n - 1)} \cdot c$$

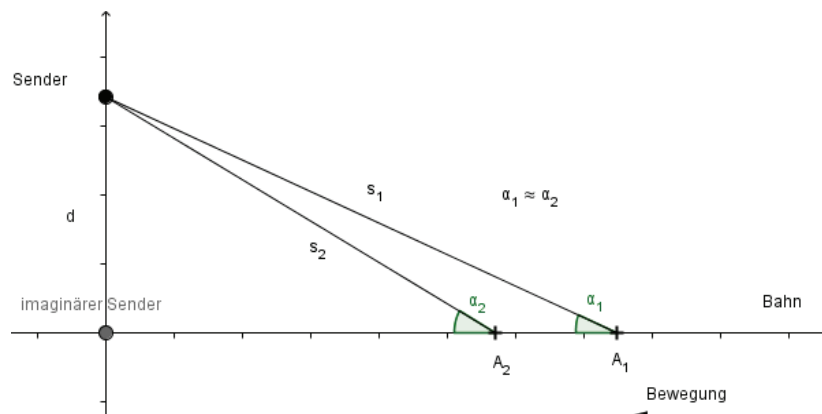
### c) Zulässiger Aufenthaltsort während der Messung

Analog zum Geschwindigkeitsbereich kann auch ein Bereich bestimmt werden, in dem sich die Wand zum Zeitpunkt 0 befinden muss, um eine korrekte Messung durchführen zu können. Dabei kommen auch die für das Geschwindigkeitsintervall relevanten Ungleichungen der Signalausendezeitpunkte zum Einsatz. Bei gegebener Geschwindigkeit  $w$  muss sich das Objekt in den unten angegebenen Grenzen befinden.

$$x_0 \leq \max \left\{ \frac{\Delta t \cdot (c - w) - (c + w) \cdot PW}{2}, \frac{\Delta t \cdot (c - w) \cdot (2 \cdot n - 1) - (c + w) \cdot PW}{2} \right\}$$

$$x_0 \geq \min \left\{ \frac{(c - w) \cdot PW}{2}, \frac{(c - w) \cdot PW}{2} - w \cdot (n - 1) \cdot \Delta t \right\}$$

### d) Erweiterung der Messsituation: Messanlage außerhalb der Bahn



In der Praxis befindet sich die Messanlage im Normalabstand  $d$  von der Bahn und die Signalausbreitungsrichtung muss mit der Bahn einen vorgeschriebenen Winkel  $\alpha$  einschließen.  $\alpha$  liegt im Intervall  $20^\circ$ - $22^\circ$ . Wir stellten fest,

dass wir die alten Formeln näherungsweise verwenden konnten, indem wir in ihnen  $c$  durch  $c \cdot \cos(\alpha)$  ersetzten. Danach leiteten wir auch die exakten Formeln zur Geschwindigkeitsbestimmung her.

## 2 Dopplereffekt: klassisch und relativistisch

### a) Galilei-Transformation

Bewegt sich ein Koordinatensystem  $S'$  in einem Koordinatensystem  $S$  mit der Geschwindigkeit  $v$ , dann werden die Koordinaten  $(x', t')$  im Koordinatensystem  $S'$  gemäß Galilei wie folgt berechnet:

$$\text{Galileitransformation: } \begin{pmatrix} x' \\ t' \end{pmatrix} = \begin{pmatrix} 1 & -v \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ t \end{pmatrix} \leftrightarrow \begin{pmatrix} x \\ t \end{pmatrix} = \begin{pmatrix} 1 & v \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x' \\ t' \end{pmatrix}$$

Man erkennt, dass  $t'_2 - t'_1 = t_2 - t_1$  ist. Dies ist in der Relativitätstheorie nicht mehr der Fall.

$$\Delta t' = \frac{c}{c-v} \cdot \Delta t$$

### b) Relativistischer Dopplereffekt

In der Relativitätstheorie muss die Galileitransformation durch die Lorentztransformation ersetzt werden:

$$\text{Lorentztransformation: } \begin{pmatrix} x' \\ t' \end{pmatrix} = \frac{1}{\sqrt{1-\left(\frac{v}{c}\right)^2}} \cdot \begin{pmatrix} 1 & -v \\ \frac{-v}{c^2} & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ t \end{pmatrix}$$

Nun erhalten wir eine andere Formel für  $\Delta t'$ :

$$\Delta t' = \sqrt{\frac{c+v}{c-v}} \cdot \Delta t$$

Für unsere Überlegungen spielt diese Formel allerdings keine Rolle, da wir alles aus einem Koordinatensystem betrachtet haben.

## 3 Berücksichtigung der Wellennatur der Signale

Durch Amplitudenmodulation können viele ähnliche Signale (z.B. Audio) deutlich unterscheidbar auf sehr hohen Trägerfrequenzen übertragen werden. Die anschließende Trennung der überlagerten Signale kann mit Hilfe der Fourier-Reihe mathematisch realisiert werden, das heißt durch numerische Integration werden die Koeffizienten berechnet, die zur Wiederherstellung der Ausgangssignale herangezogen werden.

Projekt Navigation

# Finden von optimalen Wegen in Umgebungen mit Hindernissen

Dr. Kristian Bredies

TeilnehmerInnen: Maximilian Köller, Markus Krainz, Martin Stadler, Robert Schwarzl, Florian Conrad, Klara Saller

## **PROBLEMSTELLUNG:**

Das Ziel unseres Projekts ist die Erstellung eines einheitlichen Modells für vier Problemstellungen. Diese lauten, den kürzesten bzw. den kostengünstigsten Weg durch ein Labyrinth, auf einen Berg und im Straßennetz zu finden.

Das Labyrinth besteht aus Kästchen mit vier Seiten, wobei einige dieser Seiten unpassierbar sind. Beim Besteigen eines Berges ist der kürzeste Weg nicht zugleich der leichteste, da man zusätzlich die Steigung von einem Punkt zum anderen berücksichtigen muss. Die Aufgabe im Straßennetz besteht darin, von einem Punkt zu einem anderen zu navigieren, wie es auch Navigationsgeräte im Auto und Internetseiten wie Google Maps tun.

## **ERGEBNISSE:**

### **1 Chronologische Entwicklung der Lösungsansätze**

Der Einfachheit wegen beschäftigten wir uns als erstes mit dem Labyrinth, da es sich von vornherein um ein 2D-Objekt handelt und keine Zusatzfaktoren berücksichtigt werden müssen.

#### a) Rechtsregel und CutLoops

Unsere erste Strategie beruht auf der uns bekannten Rechtsregel, die besagt, bei jeder Weggabelung den rechten Weg zu wählen bzw. sich an der rechten Wand zu halten. Der durchgangene Weg ist jedoch sehr lang und man passiert etliche Kästchen mehrmals. Man entfernt nun die Schleifen, also die Abschnitte des Weges, dessen Beginn und Ende im gleichen Kästchen liegen. Diesen Algorithmus nennen wir "CutLoops". Bei Labyrinth mit einem eindeutigen Lösungsweg ("maze") erhält man den kürzesten Weg. Bei Labyrinth mit mehreren Lösungswegen ("strange maze") ist das Ergebnis allerdings nicht zwangsläufig das gewünschte.

#### b) TAP

Ein weiterer Lösungsansatz ist der von uns entwickelte "TAP" (= Try All Possibilities) - Algorithmus. Kurz gesagt überprüft TAP systematisch alle Möglichkeiten, ans Ziel zu kommen. Je größer das Labyrinth jedoch ist, desto ineffizienter ist dieser Algorithmus.

#### c) FillUp

Parallel dazu entstand ein weiterer Algorithmus, "FillUp". Bei diesem werden Kästchen, die bereits von drei Wänden umgeben sind, gänzlich geschlossen, wodurch sich bei mehrmaliger Ausführung jede Sackgasse eines "maze" schließt und der Lösungsweg auch optisch sichtbar wird. Bei "strange mazes" führt dieser Lösungsansatz jedoch zu keinem befriedigendem Ergebnis.



#### d) Spezieller Flow

Unser "Spezieller Flow"-Algorithmus berücksichtigte als erste Strategie auch die Richtung jedes Kästchens zum Ziel. Man beginnt mit dem Kästchen vor dem Ziel, dessen Richtung man sofort eindeutig feststellen kann, und befüllt es mit der Zahl "1". Im nächsten Schritt werden alle Nachbarn dieses Kästchens mit "2" gefüllt und wiederum deren Nachbarn mit "3". "Flow" wird nun so oft wiederholt, bis auf diese Weise das Startfeld mit der Zahl "n" erreicht wird. Automatisch erhält man mit der Zahl "n" die Weglänge.

## 2 Resultate

Wir entdeckten, dass man den "Speziellen Flow" auch in ähnlicher Form auf das Berg- und auf das Straßennetzproblem umzusetzen könnte. Dafür wird aber auch ein einheitliches Modell für diese Problemstellungen benötigt.

Gegeben sei:

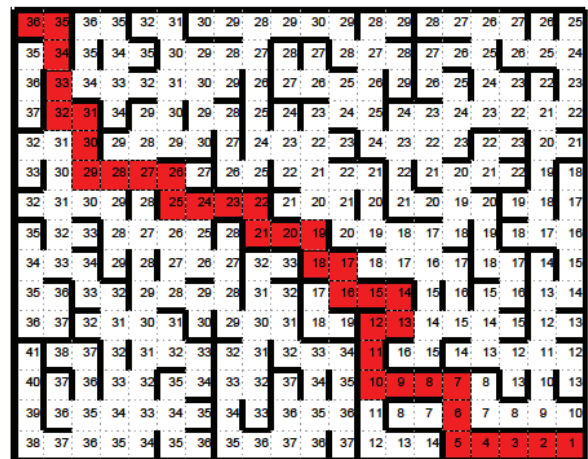
- Menge aller Punkte  $K$ ,
- Menge aller benachbarten Punkte  $E$
- Funktion  $: E \rightarrow \mathbb{R}_{\geq 0}$  gibt die Kosten von einem Punkt zum anderen an

Gesucht sei:

- Ein Weg  $W = [K_1, \dots, K_n]$   $K_1 = A$ ,  $K_n = B$ , dessen Kosten gleich das Minimum der Kosten aller Wege von A nach B ist.

#### a) Labyrinth

Das Labyrinth ist relativ einfach zu modellieren. Es ist ein 2D-Objekt und beim Begehen des benachbarten Kästchens gibt es nur zwei Möglichkeiten: Entweder der Weg ist passierbar und hat damit einen Kostenaufwand von "1" oder die Kästchen werden durch Wände getrennt womit der Aufwand "0" bzw. " $\infty$ " beträgt.

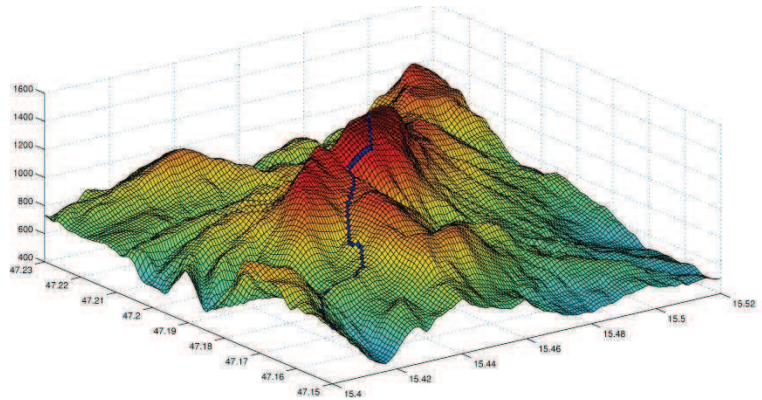


#### b) Berg

Auch den Berg kann man zu einem 2D-Objekt vereinfachen, wobei jeder Punkt zu einem Kästchen wird. Der Aufwand zwischen den Kästchen ist jetzt jedoch nicht mehr "1" oder "0", sondern die Höhenunterschiede von einem Kästchen zum anderen müssen berücksichtigt werden. Um diesen Aufwand zu berechnen erstellen wir folgende Funktion:

$$S(x) = \sqrt{1 + (xn)^2}$$

"X" beschreibt den Höhenunterschied, "n" bezeichnet den Anstrengungsfaktor, der beschreibt um wie viel anstrengender es ist, einen Meter in die Höhe als in der Ebene zurückzulegen.



### c) Straßenkarte

Die Straßenkarte erstellen wir mit gegebenen Punkten des OpenStreetMap-Projekts, wobei das gegebene Format völlig ungeeignet für MatLab ist, weshalb ein von uns entwickeltes Programm die Abstände zwischen den Straßenpunkten berechnet und sie in ein für MatLab lesbares Format bringt.

### d) Allgemeiner Flow

Für das einheitliche Modell entwickelten wir also den "Allgemeinen Flow"-Algorithmus. Bei diesem werden die Kosten zu allen benachbarten Feldern des Ausgangskästchens berechnet. Sowohl Werte als auch Felder werden in einer Liste U gespeichert, die alle "unsicheren" Werte enthält. Diese Liste wird nun geprüft und der oder die kleinsten Werte werden in eine Liste S, welche die feststehenden Werte enthalten soll, übertragen. Als nächstes werden die Nachbarn des letzten Eintrags aus der Liste S überprüft. Diese Vorgänge werden wie schon beim "Speziellen Flow" solange durchgeführt, bis man das Startkästchen erreicht hat.

## 3 Realisierung am PC

Parallel zur Findung neuer Algorithmen versuchten wir diese auch gleichzeitig auf dem Rechner zu verwirklichen. In MatLab fanden wir ein sehr mächtiges Werkzeug, das uns erlaubte die Algorithmen relativ einfach auf dem Rechner zu implementieren. Einzig bei der Effizienz stießen wir an unsere Grenzen, unter anderem weil MatLab eine Skriptsprache ist, unsere Laptops mit ihren 2 GB Arbeitsspeicher nicht in der Lage waren größere Datenmengen (zB. eine hochaufgelöste Karte des Schöckls oder eine Straßenkarte von Graz) effizient zu verarbeiten und wir uns erst mit den Feinheiten dieser Sprache vertraut machen mussten.

Projekt Dynamische Systeme  
**Gibt es Zeit-diskrete Pumpen?**  
Dr. Georg Propst

TeilnehmerInnen: Anna Eibel, Tomas Kamencek, Nadja Kravanja, Armin Shan, Philipp Schröttner, Christopher Tscherne

**PROBLEMSTELLUNG:**

Ziel unseres Projektes war es, herauszufinden, ob es zeitdiskrete Pumpen der Form

$$x_{t+1} = F(x_t) + s_t, \quad t = 0, 1, 2, 3 \dots$$

gibt.

Eine Pumpe ist so ein System genau dann, wenn es für eine periodische Störung  $s_t$  eine periodische Lösung  $x_t$  gibt, sodass für die Mittelwerte  $\bar{s}$ ,  $\bar{x}$  gilt:

$$\bar{x} \neq F(\bar{x}) + \bar{s}$$

Sowohl  $F(x)$  als auch  $s_t$  können beliebig gewählt werden.

Die Bezeichnung „Pumpe“ kommt daher, dass auch reale Pumpen ihr System durch Störungen in einem stabilen Ungleichgewicht halten können.

**ERGEBNISSE:**

**1 Affin-lineare Systeme**

Ein affin-lineares System hat die Form:  $F(x) = G(x) + d$  mit linearem  $G$ .

Behauptung: Es gibt keine affin-lineare Pumpe.

Eine Pumpe muss  $(1+n)$ -periodische Lösungen (für  $n \in \mathbb{N}^*$ ) besitzen, da man den Mittelwert bestimmen können muss, und  $(1)$ -periodische Lösungen Fixwerte sind.

Beweis, dass es keine affin-linearen Pumpen gibt, durch Annahme  $\bar{x} \neq F(\bar{x}) + \bar{s}$  und Finden eines Widerspruches:

$$\bar{x} = \frac{1}{T} \sum_{t=0}^{T-1} x_{t+1} = \frac{1}{T} \sum_{t=0}^{T-1} F(x_t) + \bar{s} = G\left(\frac{1}{T} \sum_{t=0}^{T-1} x_t\right) + \frac{1}{T} \sum_{t=0}^{T-1} d + \bar{s}$$

$$\text{also } \bar{x} = G(\bar{x}) + d + \bar{s} = F(\bar{x}) + \bar{s}$$

für  $T=(1+n)$ .

**2 Quadratische Modelle**

Versuch mit dem Modell:  $x_{t+1} = px_t(1 - x_t) + s$  für  $\{x \in \mathbb{R} | 0 \leq x \leq 1\}$ ,

also 
$$F(x) = px(1 - x)$$

a)  $s = 0$ : mathematische Analyse

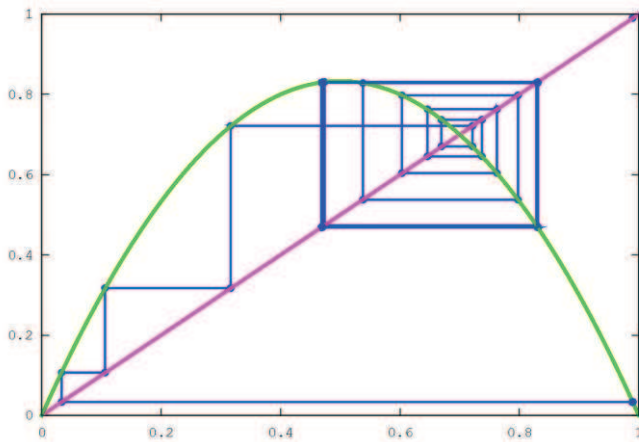
Durchführung einer Kurvendiskussion:

Hochpunkt für alle  $p$  bei  $x = \frac{1}{2}$

Die Funktion besitzt 2 Schnittpunkte mit der ersten Mediane, einen im

Koordinatenursprung, den zweiten bei  $x = 1 - \frac{1}{p}$ .

Wenn  $p > 2$ , besitzt der Schnittpunkt ( $S_1$ ) der Funktion mit der ersten Mediane einen höheren  $x$ -Wert als der Hochpunkt, was Iterationen zu einem Fixpunkt für  $2 < x < 3$  ermöglicht.



Iterationen ergeben sich durch wiederholtes Einsetzen des Funktionswertes in die Funktion.

In der Abbildung beginnt die Folge in  $x_0$  rechts unten und konvergiert gegen eine (2)-periodische Lösung (Rechteck).

**Behauptung:** Es gibt (2)-periodische Pumpen für dieses System:

**Beweis:**  $x_{t+2} = F(F(x)) = x_t$

$$F(F(x)) = p^2 x(1-x)(1-px(1-x))$$

Durch Gleichsetzen mit  $x$  ergibt sich folgende Gleichung ( $x=0$  ist nicht interessant):

$$px^3 - 2px^2 + px + x - 1 + \frac{1}{p^2} = 0$$

Mit Hilfe des Zwischenwertsatzes lässt sich zeigen:  $\exists x \in ]0, \frac{p-1}{p}[$  das diese Bedingung erfüllt. Daher muss es eine (2)-periodische Lösung  $x_1, x_2 = F(x_1)$  geben.

Dieses Modell könnte also eine Pumpe sein. Beweis durch Annahme  $\bar{x} = F(\bar{x})$  und Widerspruch:

$$\bar{x} = \frac{px_1 - px_1^2 + x_1}{2}$$

$$F(\bar{x}) = p \left( \frac{px_1 - px_1^2 + x_1}{2} \right) \left( 1 - \frac{px_1 - px_1^2 + x_1}{2} \right)$$

Durch Gleichsetzen ergibt sich eine quadratische Gleichung mit folgenden Lösungen:

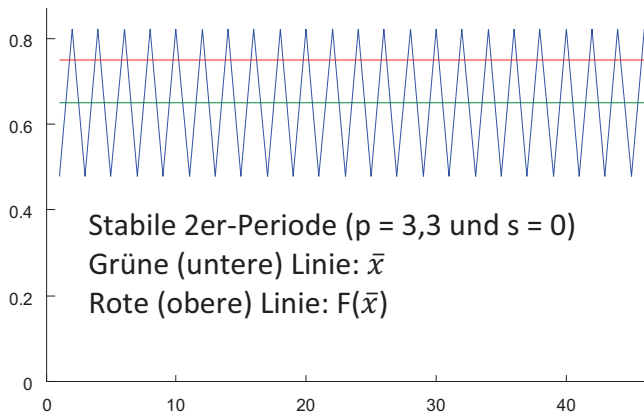
$$x_{1_1} = \frac{p-1}{p} \quad x_{1_2} = \frac{2}{p}$$

Die Lösung  $x_{1_1}$  ist nicht in  $]0, \frac{p-1}{p}[$  enthalten und  $x_{1_2}$  führt zu einem Widerspruch, da  $F(F(x_{1_2}))$ , das sich aus  $x_{1_2}$  berechnen lässt, für  $p > 0$  nicht  $x_{1_2}$  ergibt.

Bei diesem Modell handelt es sich also um eine Pumpe.

b)  $s \neq 0$ : numerische Experimente

Auch dieses Modell ist bei geeigneter Einstellung von  $p$  und  $s$  eine Pumpe, wobei  $p$ -Werte, die ohne Störung keine Pumpen waren, durch Störung zu einer Pumpe werden können und umgekehrt. Dies lässt sich gut durch numerische Berechnungen zeigen (siehe Abbildung).

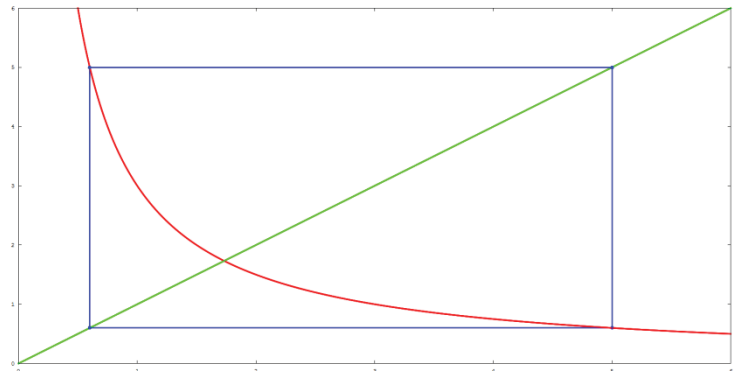


Weitere interessante numerische Experimente sind im Lang-Bericht dokumentiert.

### 3 Hyperbolische Modelle

Die Abbildung zeigt eine (2)-periodische Folge  $x_t$ , welche rechts unten bzw. links oben beginnt.

Wir versuchten analog zu den quadratischen Funktionen auch für Modelle, die durch Hyperbeln beschrieben werden, Pumpen zu finden, und zwar für  $s_t=0$ .



$$F(x_t) = \frac{p}{x_t} \text{ mit } p > 0, \quad x_{t+2} = F(F(x_t)) = \frac{px_t}{p} = x_t$$

Daher gibt es (2)-periodische Lösungen.

Beweis, dass diese Pumpen sind:

$$\bar{x} = 1/2 \left( \frac{p}{x_t} + x_t \right), \quad F(\bar{x}) = \frac{2px_t}{p + x_t^2}$$

Durch Gleichsetzen ergibt sich nur eine Lösung:  $x_{1,2} = \sqrt{p}$ . Dies ist der Schnittpunkt mit der ersten Mediane (=Fixpunkt). Das bedeutet, dass für alle anderen  $x_t \neq \sqrt{p}$  gilt:

$\bar{x} \neq F(\bar{x})$ . Also ist dieses Modell für alle  $x_0 \neq \sqrt{p}$  eine Pumpe.

### 4 Quadratwurzel-Modell

Die Funktion  $x_t = \sqrt{x_t} + s_t$  ist für  $s = 0$  keine Pumpe, da es keine periodischen Lösungen außer  $x_t=0,1$  gibt.

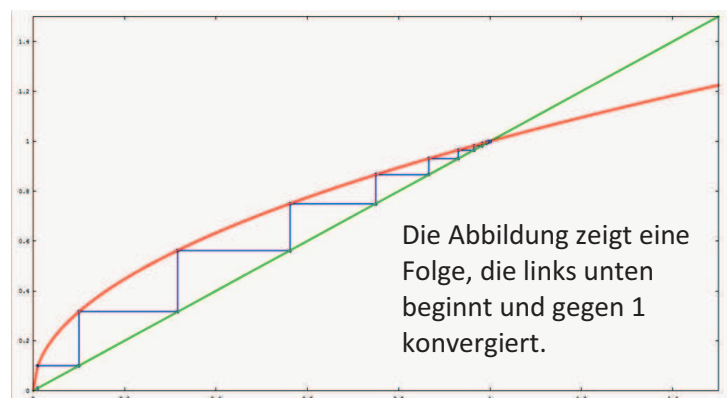
Deshalb erzeugen wir eine Störung  $s_0 = 0, s_1 \neq 0$  sodass wir eine (2)-periodische Lösung erhalten.

$$x_1 = \sqrt{x_0}, \quad x_2 = \sqrt[4]{x_0} + s_1 \text{ wobei } x_0 \neq 1 \text{ ein beliebiger Startwert ist.}$$

Durch Gleichsetzen von  $x_2$  mit  $x_0$  ergibt sich für  $s_1$ :  $s_1 = x_0 - \sqrt[4]{x_0}$

$$\bar{x} = 1/2 (x_0 + \sqrt{x_0}) \quad F(\bar{x}) = \sqrt{1/2 (x_0 + \sqrt{x_0})} \quad \bar{s} = s_1/2$$

Wegen  $\bar{x} \neq F(\bar{x}) + \bar{s}$  ist das System für alle  $x_0 \neq 0,1$  eine Pumpe.



Die Abbildung zeigt eine Folge, die links unten beginnt und gegen 1 konvergiert.

Projekt Informationstechnik

# Googles PageRank-Algorithmus

Dr. Christian Clason

Teilnehmer: Elke Schlager, Philipp Gabler, Michael Reichelt, Tristan Weniger, Lukas Grabenwarter, Konstantin Pollanz

## **PROBLEMSTELLUNG:**

Wir alle kennen Google, und wir alle benutzen es, wahrscheinlich mehrmals täglich. Aber wie oft haben Sie sich schon gefragt, warum ausgerechnet Google? Es gibt ja auch andere Suchmaschinen, die man verwenden könnte. Warum nimmt selbst Google diese Konkurrenz nicht so ernst, wie sie sein könnte? Die Antwort ist leicht: Google ist einfach besser. Um Längen. Aber wieso? Einer der Gründe, warum Google so beliebt ist, ist die Tatsache, dass man oft sofort findet, was man gesucht hat. Dies geschieht mit Hilfe eines Programms, das die Seiten mit dem gewünschten Suchbegriff nach Relevanz ordnet. Es funktioniert ziemlich gut: Die meisten Suchen enden bereits auf der ersten Seite der Google-Ergebnisse, da diese genau die richtige Information beinhalten. Unsere Aufgabe diese Woche war es, ebenfalls so einen Algorithmus zu kreieren: Den sogenannten „PageRank-Algorithmus“.

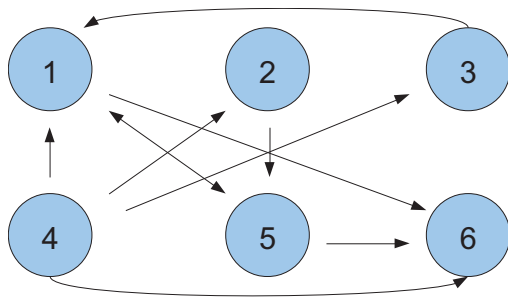
## **ERGEBNISSE:**

### **1 Das Finden eines PageRank Algorithmus**

Die erste Schwierigkeit ist es, zu verstehen, wie man diese Seiten nun nach Relevanz ordnen könnte, beziehungsweise ihre Relevanz herauszufinden - Schließlich kann es nicht die Lösung sein alle Einträge durchzulesen. Wir brauchen eine Vorgehensweise, wenn möglich eine, die nicht manipulierbar ist. Die Häufigkeit eines Suchbegriffs auf einer Seite ist zum Beispiel sehr leicht zu verfälschen. Anfangs versuchten wir, über die Verlinkungen einer Seite ihren „Wert“ herauszufinden: Je öfter eine Seite verlinkt wird, desto wertvoller erscheint sie in unserem System. Natürlich können wir hier nicht alle Links zählen, sonst wäre das Ergebnis unbrauchbar: Mehrere gleiche Links von einer Seite zu einer anderen wären zum Beispiel nichts weiter als Spam und sollten dementsprechend auch nicht bewertet werden. Eine Seite, die sich selbst verlinkt, kann man auch nicht als besser betrachten als eine, die dies nicht tut. Nachdem wir das in Betracht gezogen hatten, kamen wir zu unseren Einschränkungen, wie zum Beispiel bei mehreren gleichen Links nur einen davon zu zählen und Verlinkungen auf die eigene Seite gar nicht.

Nun konnten wir beginnen, Seiten zu bewerten. Aber wie rechnet man tatsächlich aus, wie wichtig diese Links sind? Man braucht einen Startwert, mit dem jede Seite in den Algorithmus einsteigt: In unserem Fall beträgt dieser Wert eins. Jede Seite bekommt nun pro Link, der auf sie zeigt, den Wert der verlinkenden Seite addiert. Nach mehreren Wiederholungen dieses Schrittes bekommen wir eine Rangliste, welche sich nicht mehr verändert: Unseren PageRank, zumindest unseren ersten. Wir realisierten unseren Algorithmus mit dem mathematischen Berechnungsprogramm „Octave“, das mit Matrizen arbeitet.

Deshalb mussten wir einen Weg finden, eine Link-Struktur als Matrix darzustellen und haben uns für folgenden entschieden: Wenn die Seite  $i$  auf die Seite  $j$  verweist, dann wird in der Matrix  $A$   $a_{ij}$  auf eins gesetzt. Die übrigen Werte in der Matrix  $A$  betragen null (siehe Abbildung).



**Bsp. zur Linkstruktur (bei 6 Seiten)**

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

**Matrix zum Diagramm (links)**

Wir stießen bei der Implementierung auf mehrere Probleme, wie die Tatsache, dass die Werte schnell ins Unendliche gingen. Deshalb suchten wir nach weiteren Möglichkeiten zur Sortierung. Im Endeffekt implementierten wir den bereits erwähnten Algorithmus mit kleinen Änderungen am PC. Da jedoch die übergroßen Werte für Computer nicht unterscheidbar waren, konnten wir die Rangordnung der Seiten mit diesem Algorithmus nicht feststellen.

Aber nur aufgrund dieses Wertes kann man Seiten ohnehin nicht präzise ordnen. Eine Seite, die zwar einen unglaublich guten Inhalt aufweist, aber noch nicht verlinkt wurde, wäre durch diesen Algorithmus sehr benachteiligt. Das allein kann also nicht die Lösung unseres Problems sein, aber ein Teil unserer Antwort ist es: Wir sehen diesen Wert proportional zu anderen Faktoren, zum Beispiel der Häufigkeit des Suchwortes im Text, oder auch der Beliebtheit der gesamten Domain. Als dritten Faktor zählen wir die sogenannte Klick-Statistik zu der bis jetzt errechneten Wertung dazu. Diese bestimmt, wie oft die Seiten über unsere Suchmaschine in der Vergangenheit besucht wurden und bewertet sie dementsprechend.

## 2 Implementierung und Optimierung des Algorithmus

Es scheint bereits ein komplexes Problem zu sein, nur einen funktionierenden Algorithmus zu finden, aber das war noch nicht alles: Nicht jeder Mensch hat endlos viel Zeit um auf seine Suchergebnisse zu warten, doch bis auf die Möglichkeit, einige Schleifen durch einfachere Strukturen zu ersetzen, kann man ein Programm kaum schneller machen, ohne dabei Genauigkeit einzubüßen. Wir mussten also einen Kompromiss zwischen Präzision und Geschwindigkeit finden. Außerdem mussten wir darauf achten, gewisse Strukturen vermeiden. Ein Programm, dessen Laufzeit kubisch mit der Anzahl der zu untersuchenden Seiten ansteigt wäre zum Beispiel unbrauchbar, da es zu viel Zeit in Anspruch nähme. Zu ungenau darf das Programm aber auch nicht werden. Ein weiteres Problem neben den Parametern waren die Werte im Allgemeinen, da diese schnell gegen Unendlich oder Null konvergierten. Im Endeffekt entschieden wir uns also dafür, den Wert einer Seite als die Wahrscheinlichkeit zu interpretieren, mit der ein Surfer diese Seite besucht, wobei sich die Einzelwahrscheinlichkeiten auf eins summieren. Diese Tatsache hilft auch, die verschiedenen Seitenwerte zu vergleichen. Ein weiteres Problem ließen komplett voneinander getrennte Subnetzwerke entstehen. Sie stellten unseren Algorithmus vor schwer lösbare Probleme, weil sich die Verhältnisse der Wertigkeiten dieser Netze nicht bestimmen ließen. Die Lösung dazu war eine imaginäre Seite, die sowohl von allen Seiten verlinkt wird als auch auf alle verweist. Sobald wir diese inkludierten, wurde es möglich, alle Seiten zufriedenstellend zu bewerten, auch wenn unsere imaginäre Seite nie außerhalb des Programmcodes vorkommt, und deswegen auch die Ergebnisse nicht wirklich beeinträchtigt.

$$\mathbf{A} = (a_{ij}) \in \mathbb{R}^{n \times n}$$

$$\mathbf{s} = (s_i) \quad \text{mit} \quad s_i = \left( \left( \sum_{j=1}^n a_{ij} \right) + 1 \right)^{-1} \quad \text{für} \quad 1 \leq i \leq n$$

$$\tilde{\mathbf{A}} = (\tilde{a}_{ij}) = (a_{ji}s_j) \quad \text{für} \quad 1 \leq i, j \leq n$$

$$\mathbf{x}^1 = \frac{1}{n} \mathbf{e} \quad \text{mit} \quad \mathbf{e} = (1, \dots, 1)^T \in \mathbb{R}^n$$

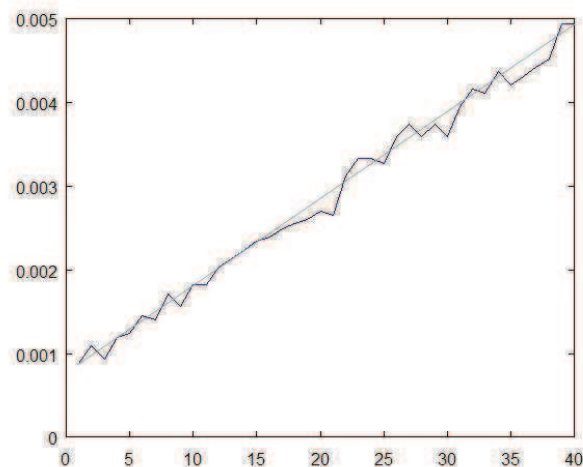
$$\mathbf{x}^{k+1} = \tilde{\mathbf{A}} \mathbf{x}^k + \frac{\mathbf{s}^T \mathbf{x}^k}{n} \mathbf{e} \quad \text{mit} \quad k \rightarrow \infty$$

Unser Endalgorithmus

### 3 Vergleich unseres Modells mit dem Googles

Google verwendet einen PageRank-Algorithmus, der unserem sehr ähnelt. Es gibt jedoch einen großen Unterschied. Während wir das Problem der Subnetze wie beschrieben durch eine imaginäre Seite lösen, geht Google einen etwas anderen Weg. Es wird eine Wahrscheinlichkeit von 0,15 angenommen, dass der Zufallssurfer, der normalerweise zufällig auf Links klickt, zu einer beliebigen anderen Seite springt unabhängig davon, ob zwischen den beiden Seiten ein Link besteht.

Dabei ist die Wahrscheinlichkeit von 0,15 ein willkürlich gewählter Wert. Würde man in Googles Modell diesen Wert auf 0,43 setzen, würden sich für die einzelnen Seiten sehr ähnliche Wertigkeiten ergeben wie bei unserem System.



Laufzeit als Funktion der Seitenanzahl (in Hunderterschritten)



Projekt Politikwissenschaft  
**Entwicklung eines Wahlsystems**  
Dr. Stephen Keeling

Teilnehmer: Thessa Hinteregger, Simeon Kanya, Benedikt Maderbacher, Sarah Österreicher, Martin Schwarzl, Julian Greitler

**PROBLEMSTELLUNG:**

Unser Ziel war die Entwicklung eines geeigneten Wahlmodells, das die folgenden Fragen behandelt: Was ist Gemeinwohl überhaupt? Wie misst man das Gemeinwohl? Würden die Bürger ihre Präferenzen ehrlich bekannt geben oder eher ein Wahlsystem manipulieren? Laut dem Satz von Arrow kann es keine gemeinsame Präferenzenliste geben ohne natürliche Gerechtigkeitsbedingungen zu verletzen. Trotzdem müssen Entscheidungen getroffen werden. Kann für solch eine gemeinsame Entscheidung eine Zielfunktion gefunden werden beziehungsweise kann es ein stabiles Ergebnis geben? Um das zu untersuchen sollen neue Wahlsysteme gefunden werden. Außerdem sollen die Veränderungen dieser Faktoren bei mehreren Wahlgängen untersucht werden.

**BERICHT:**

Man betrachte das Ägyptenproblem: In einer Gruppe soll eine Entscheidung für ein Reiseziel getroffen werden. Eine geeinte Minderheit will nach Ägypten, während die Übrigen sich jeweils für ein anderes Reiseziel aussprechen. Die endgültige Entscheidung fällt jedoch auf Ägypten.

Damit wir uns besser in die Situation des Ägypten-Problems hineinversetzen konnten, entschieden wir uns kurzfristig eine Abstimmung zwischen sechs verschiedenen Urlaubsdestinationen durchzuführen. Folglich kamen verschiedene Fragen auf, bezüglich Manipulation, Bewertungsmethoden, Stabilität, et cetera.

Um unsere Theorien in größerem Rahmen zu testen, machten wir eine Umfrage bezüglich der Freizeitaktivitäten auf der Modellierungswoche unter allen Teilnehmern. Die Befragten konnten die Möglichkeiten mit Werten zwischen 0 und 10 bewerten, wobei höchstens zwei Werte im ablehnenden Bereich zwischen 0 und 4 liegen durften.

Anhand der Daten versuchten wir ein erstes Modell für ein Wahlsystem aufzustellen. Wir waren uns einig, dass das arithmetische Mittel und die Unzufriedenheit pro Person wichtig waren um ein Maß für das Gemeinwohl zu finden und so stellten wir folgende Formel auf:

$$a = \frac{\bar{x}}{b + 1}$$

*a....gesuchter Wert  
b....Unzufriedenheit pro Person*

Das nächste Modell basiert darauf, dass eine gemeinsame Entscheidung durch Gruppenbildung gefunden wird. Wir gehen davon aus, dass hier auch die Dominanz eine wichtige Rolle spielt, was de facto später auf Manipulation umgelegt werden kann. Im zweiten Wahlgang wäre die neue Meinung M dann:

$$M(i) = \frac{\sum_i^j D(i) * Malt(i)}{\prod_i^j D(i)}$$

*D.... Dominanz (Zufallsvariable)*

*M....Meinung nach n Wahlgängen  
Malt...Meinung vor dem n-ten Wahlgang*

Wenn wir keine Dominanz beziehungsweise Manipulation haben, weil wir diese ja minimieren wollen, so ergibt sich als Endergebnis das arithmetische Mittel der Anfangsmeinungen, woraus wir schließen, dass mehrere Wahlgänge in diesem Fall nicht sinnvoll sind.

Ein weiteres Modell ist ein präferenzbezogenes. Wie der Name schon sagt, wird auf Basis der Präferenzen ein Modell erstellt, welches einen gemeinsamen Präferenzpunkt berechnet. Im Vergleich zu einem System mit einem Wahlgang, schreiben wir zusätzlich zwei Programme, die sich mehrerer Wahlgänge bedienen. Der Unterschied der Mehrwahlsysteme liegt in der Taktik der Spieler. Beim ersten Modell handeln sie strikt nach einem Muster, welches stures Präferenzverhalten implementiert. Das zweite System agiert intelligenter. Es untersucht eine von der Anzahl der Wahlmöglichkeit abhängig Quantität von Punkten auf den höchsten Gewinn und wählt die Präferenzen dieses Punktes für den nächsten Wahlgang aus. Aus der Gegenüberstellung der Methoden resultiert ein klarer Vorteil für das erste Modell, da mehrere Wahlgänge zwar zu einem Gleichgewicht führen, jedoch das Wahlergebnis erheblich verfälschen. Dies trifft auf ein System mit einem Wahlgang nicht zu. Eine andere interessante Beobachtung konnten wir bei der Zielfunktion feststellen, welche unstetig ist. Sobald der allgemeine Präferenzpunkt über eine Schwerlinie des Polygons tritt, springt der Ausgabewert der Funktion nach oben beziehungsweise unten.

Im letzten Modell stellen wir die Meinungen der einzelnen Personen in einem n-Simplex da, diese Figur hat gleich viele Eckpunkte wie Auswahlmöglichkeiten zur Verfügung stehen, ist aber nicht zweidimensional sondern hat genau eine Dimension weniger als Auswahlmöglichkeiten. Bei drei Möglichkeiten entsteht also ein Dreieck. Für unser Modell haben wir festgelegt, dass jede Person Resistenzen für jede Möglichkeit angeben muss und sich nur auf der Kante bewegen darf das seinem Favoriten gegenüberliegt. Der Sieger in unserer Simulation ist der Eckpunkt, der den größten Abstand zum Durchschnittspunkt aller Meinungen aufweist. Wenn man den Wahldurchgang unter der Annahme wiederholt, dass jeder Wähler seinen Gewinn maximieren will, streben alle Punkte auf die Mittelpunkte der Seiten zu.

Im Laufe der Modellierungswoche kamen wir zu dem Schluss, dass das Gemeinwohl die höchstmögliche Zufriedenheit bei möglichst geringer Unzufriedenheit darstellt. Unsere Untersuchungen ergaben, dass kein Wahlsystem alle Bedingungen des Satzes von Arrow erfüllen kann und bestätigten ihn damit. Wir stellten fest, dass es bei mehreren Wahlgängen zwar ein stabiles Ergebnis gibt, dieses jedoch verfälscht ist. Daher sind wir der Meinung, dass ein ideales Wahlsystem aus nur einem Wahlgang bestehen soll und die Bewertungen gewichtet sein sollen, um größtmögliche Genauigkeit zu gewährleisten. Als bestes Modell wählten wir folgendes aus:

$$a = \frac{\frac{1}{n} * \sum_{i=1}^n x_i}{\frac{1}{n} * \sum_{i=1}^m x_i + 1}$$

*a.....gesuchter Wert  
n....Anzahl der Wähler  
m.....Anzahl der unzufriedenen Wähler*