

Darstellung von Differentialgleichungen mit einem zellulären Automaten

Detlev Ziereisen, Florian Lüthi, i10b

29. April 2012

1 Einleitung

Diese Projektarbeit (bestehend aus diesem Dokument sowie der zugehörigen Software **Tusk**) hat zum Ziel, partielle Differentialgleichungen aus dem Umfeld der physikalischen Simulationen mit einem zellulären Automaten zu berechnen sowie darzustellen. Es soll gezeigt werden, dass dies unter den konzeptionellen Restriktionen der Theorie der zellulären Automaten möglich ist, insbesondere soll die Annahme Konrad Zuses, dass die Naturgesetze diskreten Regeln folgten und das gesamte Geschehen im Universum das Ergebnis der Arbeit eines gigantischen zellulären Automaten sei [7, 9], zumindest für den Scope dieser Arbeit bestätigt werden. Ausserdem soll gezeigt werden, dass die intuitiv erwarteten Resultate in sinnvoller Rechenzeit dargestellt werden können.

Diese Projektarbeit wurde im Rahmen des Kurses Softwareprojekt 2 an der Hochschule für Technik Zürich durchgeführt. Der Quellcode sowie auch dieses Dokument sind Online auf <http://www.github.com/foyan/Tusk> zu finden.

Der Autoren Dank geht (in alphabetischer Reihenfolge der Vornamen) an Albert Heuberger für geballte mathematische Kompetenz gepaart mit unendlicher Geduld, an Lukas Eppler für die ständige Begleitung, sowie an Philippe Nahlik für die tolle Kursführung.

2 Zelluläre Automaten

Ein zellulärer Automat ist eine regelmäßige Annordnung von Zellen. Jede Zelle kann eine endliche Zahl von Werten / Zuständen annehmen und hat eine begrenzte Zahl von Nachbarzellen, die sie beeinflussen können. Das Muster des gesamten zellulären Automaten ändert sich in einzelnen Schritten, die durch eine Reihe von Übergangsregeln bestimmt werden, die für alle Zellen gelten.[1]

Also:

Definition 1 (Zellulärer Automat). Ein zellulärer Automat ist durch folgende Eigenschaften festgelegt:

- einen Zellarraum R ,
- eine endliche Nachbarschaft N , wobei $\forall r \in R (N_r \subset R)$,
- eine Zustandsmenge Q ,
- eine Überföhrungsfunktion $\delta : Q^{|N|+1} \mapsto Q$.

Die Zustandsübergänge erfolgen für alle Zellen nach derselben Überföhrungsfunktion und gleichzeitig. Die Zellzustände können wie die Zeitschritte diskret sein. [7]

Bemerkung. Aus dieser Definition folgt unmittelbar, dass der neue Zustand einer Zelle nur vom momentanen Zustand dieser Zelle sowie den Zuständen der Nachbarzellen abhängig sein kann.

Für zweidimensional organisierte R sind zwei Arten von Nachbarschaft üblich:

Von-Neumann-Nachbarschaft Die Nachbarschaft besteht jeweils aus den vier geographisch nächsten Nachbarzellen: $N_{i,j} = \{R_{i,j-1}, R_{i,j+1}, R_{i-1,j}, R_{i+1,j}\}$

Moore-Nachbarschaft Die Nachbarschaft besteht jeweils aus allen Zellen der Von-Neumann-Nachbarschaft sowie zusätzlich der diagonalen Nachbarzellen:

$$N_{i,j} = \{R_{i,j-1}, R_{i,j+1}, R_{i-1,j}, R_{i+1,j}, R_{i-1,j-1}, R_{i-1,j+1}, R_{i+1,j-1}, R_{i+1,j+1}\} \text{ [3]}$$

Beispiel (Wolfram's eindimensionales Universum). Stephen Wolfram¹ definiert in [8] und in etlichen Arbeiten aus der Mitte der 1980er-Jahre einen parametrierbaren zellulären Automaten, der nur aus einer einzigen Raumdimension besteht[6]. In seiner einfachsten Ausprägung ist die Nachbarschaft N_i definiert als $\{R_{i-1}, R_{i+1}\}$, und jede Zelle kann genau zwei Zustände annehmen (tot und lebendig bzw. 0 und 1). Damit ist

$$\delta : \{0, 1\}^{|N_{i-1}, R_{i+1}|+1} \mapsto \{0, 1\} = \{0, 1\}^3 \mapsto \{0, 1\},$$

ergo existieren 8 mögliche Zustandsänderungen. Ein Beispiel:

alte $R_{i-1}R_iR_{i+1}$	111	110	101	100	011	010	001	000
neues R_i	0	1	1	0	1	1	1	0

Tabelle 1: Wolfram-Konfiguration 110

Man stellt nun fest, dass bei 8 möglichen Zustandsänderungen für die Anzahl der möglichen Konfigurationen gilt:

$$|K| = |Q|^{|dom(\delta)|} = 2^8 = 256.$$

Jeder dieser Konfigurationen kann eine natürliche Zahl $\{0, 1, 2, \dots, 255\}$ zugeordnet werden, indem die neuen R_i wie oben tabelliert und pro Zeile als binäre Zahl aufgefasst werden [2]:

¹Stephen Wolfram (* 29. August 1959), britischer Physiker und Mathematiker, Schöpfer der Software Mathematica sowie der Suchmaschine Wolfram Alpha [6]

111	110	101	100	011	010	001	000	Zahl
0	0	0	0	0	0	0	0	00000000 _b = 0 _d
0	0	0	0	0	0	0	1	00000001 _b = 1 _d
0	0	0	0	0	0	1	0	00000010 _b = 2 _d
⋮								
0	1	1	0	1	1	1	0	01101110 _b = 110 _d
⋮								
1	1	1	1	1	1	1	1	11111111 _b = 255 _d

Tabelle 2: Alle Wolfram-Konfigurationen

Dadurch ist es möglich, sämtliche eindimensionalen Wolfram-Universen durch eine einzige Zahl zu identifizieren.

Als besonders spannend hat sich die in Tabelle 1 dargestellte Konfiguration 110 erwiesen, weil sie die Eigenschaft hat, ein Turing-vollständiges System zu sein [2, 8]. Dadurch ist sie die Konfiguration einer universellen Turingmaschine, die mit nur 2 Zuständen und 5 Symbolen umgesetzt werden kann – somit hat die Wolfram-Konfiguration 110 als Turingmaschine einen Umfang von $2 \cdot 5 = 10$ und zählt damit zu den kleinsten bis dato bekannten Turingmaschinen [2].

Beispiel (Game of Life). Das von Conway² 1970 entworfene Game of Life ist eine bis heute populäre Umsetzung der Automatentheorie und insbesondere der Idee der zellulären Automaten [4].

Der ursprüngliche Entwurf befindet sich in einem zweidimensionalen R unter Verwendung der Moore-Nachbarschaft. Die Zellen können zwei mögliche Zustände $\{q_{\text{lebend}}, q_{\text{tot}}\}$ annehmen, und die Übergangsfunktion ist definiert als:

$$\delta(r, N) = \begin{cases} q_{\text{lebend}} & (r = q_{\text{tot}} \wedge \varsigma(q_{\text{lebend}}, N) = 3) \\ q_{\text{tot}} & (r = q_{\text{lebend}} \wedge \varsigma(q_{\text{lebend}}, N) < 2) \\ q_{\text{lebend}} & (r = q_{\text{lebend}} \wedge 2 \leq \varsigma(q_{\text{lebend}}, N) \leq 3) \\ q_{\text{tot}} & (r = q_{\text{lebend}} \wedge \varsigma(q_{\text{lebend}}, N) > 3) \\ q_{\text{tot}} & (\text{sonst}) \end{cases}$$

unter Zuhilfenahme der Statuszählfunktion

$$\varsigma(q, N) = \sum_{i=1}^8 \begin{cases} 1 & (N_i = q) \\ 0 & (N_i \neq q) \end{cases}$$

²John Horton Conway (* 26. Dezember 1937), englischer Mathematiker[5]

3 Vom zellulären Automaten zur Differentialgleichung

Die Berechnung und Darstellung physikalischer Begebenheiten (allgemein ausgedrückt durch folgende partielle Differentialgleichung, mit $u : u(\vec{x}, t)$)

$$k_n \frac{\partial^n u}{\partial t^n} + k_{n-1} \frac{\partial^{n-1} u}{\partial t^{n-1}} + \dots + k_1 \frac{\partial u}{\partial t} + k_0 = \frac{\partial^n u}{\partial \vec{x}^n} + \frac{\partial^{n-1} u}{\partial \vec{x}^{n-1}} + \dots + \frac{\partial u}{\partial \vec{x}}$$

mit zellulären Automaten kann durchgeführt werden, indem folgendes getan wird:

- R entspricht einer sinnvollen (groben) Diskretisierung der örtlichen Variablen \vec{x} in einer, zwei oder drei Dimensionen
- Jede Zelle in R ist ein Tupel (Q, D) mit Q als einer Menge von berechnungsfernen Zustandsinformationen und den Differentialen nach der Zeit $D = \left(u, \frac{\partial u}{\partial t}, \frac{\partial^2 u}{\partial t^2}, \dots, \frac{\partial^n u}{\partial t^n}\right) \in \mathbb{R}^n$
- Eine neue Generation entspricht jeweils der fortgelaufenen Zeit ∂t , welche sehr fein diskretisiert werden muss
- In der Übergangsfunktion δ steckt die eigentliche Differentialgleichung. In der Regel verändert sie nur die Elemente von D .

4 δ der Wellengleichung

Ausgehend von der homogenen Wellengleichung

$$\frac{\partial^2 u}{\partial t^2} = k \cdot \frac{\partial^2 u}{\partial \vec{x}^2}$$

und folgenden Zusammenhängen:

$$\begin{array}{ccc} u_i & & u_i(t + \Delta t) = u_i(t) + \frac{\partial u_i(t + \Delta t)}{\partial t} \Delta t \\ \downarrow & & \uparrow \\ \frac{\partial u_i}{\partial \vec{x}} = u_i - \sum u_{i-1} & & \frac{\partial u_i(t + \Delta t)}{\partial t} = \frac{\partial u_i(t)}{\partial t} + \frac{\partial^2 u}{\partial t^2} \Delta t \\ \downarrow & & \uparrow \\ \frac{\partial^2 u_i}{\partial \vec{x}^2} = -2u_i + \sum u_{i-1} + \sum u_{i+1} & \xrightarrow{\frac{\partial^2 u}{\partial t^2} = k \cdot \frac{\partial^2 u}{\partial \vec{x}^2}} & \frac{\partial^2 u}{\partial t^2} \end{array}$$

sowie unter der Annahme, dass die Richtung der Nachbarzellen bekannt sind (d.h. jeweils der Vorgänger beziehungsweise der Nachfolger innerhalb einer Dimension bekannt sind und deswegen N auch ausgedrückt werden kann als:)

$$N^* = \left(\begin{array}{cccc} \text{1. Dimension} & \text{2. Dimension} & \dots & \frac{1}{2}|N|\text{te Dimension} \\ \overbrace{n_{1,1} \in N} & \overbrace{n_{2,1} \in N} & \dots & \overbrace{n_{\frac{1}{2}|N|,1} \in N} & \text{Vorgänger} \\ n_{1,2} \in N & n_{2,2} \in N & \dots & n_{\frac{1}{2}|N|,2} \in N & \text{Nachfolger} \end{array} \right)$$

kann die Übergangsfunktion δ für die Wellengleichung folgendermassen formuliert werden:

$$\delta(r, N^*) =$$

5 δ der Ente

Zu jeder ernstzunehmenden Simulation von Wasser gehören sich darin (aktiv oder passiv) bewegende Objekte – im Idealfall sind das Enten. Die Differentialgleichung der Badeente im Wasser lautet:

$$\frac{\partial \vec{x}}{\partial t} = k \cdot \frac{\partial u}{\partial \vec{x}}$$

Sprich: der Geschwindigkeitsvektor im \vec{x} -Vektorfeld (also die Ableitung nach der Zeit t) ist abhängig von den Steigungen (also der Differenzen der Höhe u) entlang aller Dimensionen.

Speziell dabei ist noch, dass so eine Ente natürlich nur in endlich vielen Zellen vorkommen kann. Da die \vec{x} -Komponente

6 Numerische Verfahren

7 Bemerkungen zur Implementation

7.1 Wahl des Umfelds

Die Wahl des Umfelds fiel auf eine Kombination aus JavaScript und dem HTML5-**canvas**-Element für die grafische Ausgabe. Die hauptsächlichen Gründe dafür sind die einfache Umsetzung sowie die sehr gute Portabilität, vorallem auch auf portable Devices mit Touch-Bedienung. Dadurch kommt der primäre Use Case der Software (Bewegung einer Flüssigkeit mittels dem Pointing Input) besonders gut zur Geltung, weil der Input eben mit dem Finger vorgenommen werden kann.

Bedenken wurden vorgängig und während der ersten Implementierungsphase vor allem in Bezug auf die Perfomance geäussert, konnten aber im Laufe des Projekts zerstreut werden, auch weil gezeigt werden konnte, dass Optimierungen möglich sind (paralleles Berechnen, Umstellung von Vektorgrafik-Operationen auf Bitmap-Manipulationen).

Als Alternativen waren das Game-Framework XNA sowie die 3D-LED-Installation in der Haupthalle des Zürcher Hauptbahnhofs in Betracht gezogen worden.

8 Schlussfolgerung

Literaturverzeichnis

- [1] Hans-Georg Beckmann. Zelluläre Automaten. <http://www.vlin.de/material/ZAutomaten.pdf>, 2003. [Online; Stand 28. April 2012].

- [2] André Betz. *Das eindimensionale Universum. Einführung in ein informationstheoretisches Weltbild*. 1 edition, 2003.
- [3] Stefan Baur und Markus Hanselmann. Zelluläre Automaten. http://www5.in.tum.de/FA/_2005/K5/Schurr/11_CA.PDF, 2005. [Online; Stand 28. April 2012].
- [4] Wikipedia. Conways Spiel des Lebens — Wikipedia, Die freie Enzyklopädie. http://de.wikipedia.org/w/index.php?title=Conways_Spiel_des_Lebens&oldid=100624377, 2012. [Online; Stand 28. April 2012].
- [5] Wikipedia. John Horton Conway — Wikipedia, Die freie Enzyklopädie. http://de.wikipedia.org/w/index.php?title=John_Horton_Conway&oldid=101731222, 2012. [Online; Stand 28. April 2012].
- [6] Wikipedia. Stephen Wolfram — Wikipedia, Die freie Enzyklopädie. http://de.wikipedia.org/w/index.php?title=Stephen_Wolfram&oldid=99357821, 2012. [Online; Stand 28. April 2012].
- [7] Wikipedia. Zellulärer Automat — Wikipedia, Die freie Enzyklopädie. http://de.wikipedia.org/w/index.php?title=Zellul%C3%A4rer_Automat&oldid=100860810, 2012. [Online; Stand 28. April 2012].
- [8] Stephen Wolfram. *A New Kind of Science*. Wolfram Media, 1 edition, 5 2002.
- [9] Konrad Zuse. *Rechnender Raum*. 1969.