

12 Diskretisierung kontinuierlicher Systeme

Zum Thema dieses Kapitels gibt es eine Fülle von Literatur. Hier werden nur ganz einfache und grundlegende Aspekte zur Sprache kommen. Als weitergehende Einführung erscheint uns zum Beispiel das Buch von Fletcher 1991 sehr nützlich.

12.1 SATZ. (TAYLOR) Es sei f eine reelle Funktion einer Veränderlichen, die auf einem Intervall $I_\alpha =]x_0 - \alpha; x_0 + \alpha[$ mindestens $(n + 1)$ -mal stetig differenzierbar ist. Dann gibt es für jedes $x \in I_\alpha$ ein $\theta \in [0; 1]$, so dass

$$f(x) = \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k + R_n(x, x_0, \theta)$$

mit

$$R_n(x, x_0, \theta) = \frac{f^{(n+1)}(x_0 + \theta(x - x_0))}{(n + 1)!} (x - x_0)^{n+1}.$$

12.2 KOROLLAR. (TAYLORENTWICKLUNG) Ist f auf einem Intervall $I_\alpha =]x_0 - \alpha; x_0 + \alpha[$ unendlich oft differenzierbar und gilt für alle $x \in I_\alpha$ und alle $\theta \in [0; 1]$

$$\lim_{n \rightarrow \infty} R_n(x, x_0, \theta) = 0,$$

dann gilt für alle $x \in I_\alpha$

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k$$

bzw. für $\Delta x < \alpha$:

$$f(x_0 + \Delta x) = \sum_{k=0}^{\infty} \frac{(\Delta x)^k}{k!} f^{(k)}(x_0) = f(x_0) + \Delta x f'(x_0) + \frac{(\Delta x)^2}{2} f''(x_0) + \dots$$

Dies lässt sich auch in einer Art Operatorschreibweise angeben als

$$f(x_0 + \Delta x) = \sum_{k=0}^{\infty} \frac{1}{k!} \left(\left(\Delta x \frac{d}{dx} \right)^k f \right) (x_0).$$

Dabei ist $\frac{d}{dx}$ der Operator „Funktion ableiten“, eine Konstante c ist der Operator „Funktion mit c malnehmen“ und die „Multiplikation“ von Operatoren bedeutet ihre Hintereinanderausführung. Die „Addition“ von Operatoren, die wir auch gleich noch verwenden werden, sei „punktweise“ definiert.

Analog zu Korollar 1.2 gilt für zwei- und mehrstellige Funktionen:

12.3 SATZ. Ist $f(x, y)$ auf $I_\alpha = \{(x, y) | (x - x_0)^2 + (y - y_0)^2 < \alpha\}$ unendlich oft stetig differenzierbar, dann gilt für alle $(\Delta x, \Delta y)$ mit $(\Delta x)^2 + (\Delta y)^2 < \alpha$:

$$\begin{aligned} f(x_0 + \Delta x, y_0 + \Delta y) &= \sum_{k=0}^{\infty} \frac{1}{k!} \left(\left(\Delta x \frac{\partial}{\partial x} + \Delta y \frac{\partial}{\partial y} \right)^k f \right) (x_0, y_0) \\ &= \sum_{r,s \geq 0} \frac{(\Delta x)^r (\Delta y)^s}{r!s!} \frac{\partial^{r+s} f}{\partial x^r \partial y^s} (x_0, y_0). \end{aligned}$$

12.4 Insbesondere erhält man zum Beispiel für $\Delta y = 0$:

$$f(x_0 + \Delta x, y_0) = \sum_{k=0}^{\infty} \frac{(\Delta x)^k}{k!} \frac{\partial^k f}{\partial x^k}(x_0, y_0).$$

Analog für $\Delta x = 0$.

12.5 DEFINITION (NABLA-OPERATOR)

$$\nabla := \left(\frac{\partial}{\partial x} \quad \frac{\partial}{\partial y} \right)$$

Die Anwendung eines Vektors von Operatoren auf eine Funktion f ergebe den Vektor der Funktionen, die sich durch Anwendung jedes Operators einzeln auf f ergeben.

Außerdem wird gelegentlich das Skalarprodukt $\nabla^2 = \nabla \cdot \nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ benutzt werden.

12.6 Mit Hilfe des Nablaoperators lässt sich die Formel in Satz 1.3 auch wie folgt schreiben, wobei $(x_0, y_0) = \mathbf{x}$ gesetzt wurde und $(\Delta x, \Delta y) = \Delta \mathbf{x}$:

$$\begin{aligned} f(\mathbf{x} + \Delta \mathbf{x}) &= \sum_{k=0}^{\infty} \frac{1}{k!} \left((\Delta \mathbf{x} \cdot \nabla)^k f \right) (\mathbf{x}) \\ &= f(\mathbf{x}) + (\Delta \mathbf{x} \cdot \nabla f)(\mathbf{x}) + (\Delta \mathbf{x} \cdot \nabla f)^2(\mathbf{x}) + \sum_{k=3}^{\infty} \frac{1}{k!} \left((\Delta \mathbf{x} \cdot \nabla)^k f \right) (\mathbf{x}) \\ &= f(\mathbf{x}) + \Delta x \frac{\partial f}{\partial x}(\mathbf{x}) + \Delta y \frac{\partial f}{\partial y}(\mathbf{x}) + \frac{(\Delta x)^2}{2} \frac{\partial^2 f}{\partial x^2}(\mathbf{x}) + \Delta x \Delta y \frac{\partial^2 f}{\partial x \partial y}(\mathbf{x}) + \frac{(\Delta y)^2}{2} \frac{\partial^2 f}{\partial y^2}(\mathbf{x}) \\ &\quad + \sum_{k=3}^{\infty} \frac{1}{k!} \left((\Delta \mathbf{x} \cdot \nabla)^k f \right) (\mathbf{x}) \end{aligned} \quad (12.1)$$

Im folgenden werden wir immer Funktionen $f(t, x, y)$ betrachten, von denen wir annehmen, dass sie in der gerade betrachteten Umgebung analytisch sind. Unsere Vorstellung ist die zeitlich variierender Funktionen, die zum Beispiel für die Punkte der Ebene \mathbb{R}^2 Werte vorschreiben.

Um eine kompakte Schreibweise zu haben, definieren wir noch die folgenden Abkürzungen:

12.7 DEFINITION (DISKRETE OPERATOREN) Ein Δt , ein Δx und ein Δy seien fixiert.

$$\begin{aligned} (Tf)(t, x, y) &:= f(t + \Delta t, x, y) & (T^-f)(t, x, y) &:= f(t - \Delta t, x, y) \\ (Xf)(t, x, y) &:= f(t, x + \Delta x, y) & (X^-f)(t, x, y) &:= f(t, x - \Delta x, y) \\ (Yf)(t, x, y) &:= f(t, x, y + \Delta y) & (Y^-f)(t, x, y) &:= f(t, x, y - \Delta y) \\ (If)(t, x, y) &:= f(t, x, y) \end{aligned}$$

Eine (wenngleich auch kritisierbare und sicher nicht die einzige) Möglichkeit, nun Zellularautomaten anzuwenden, besteht darin, einige der Werte $f(t, x, y)$ zu einem festen Zeitpunkt mit den Zuständen der Zellen eines zweidimensionalen Zellularautomaten in Verbindung zu bringen. Dazu seien Werte $t_0, x_0, y_0, \Delta t, \Delta x$ und Δy fixiert und eine Funktion g wie folgt definiert:

$$\begin{aligned} g : \mathbb{N}_0 \times \mathbb{Z} \times \mathbb{Z} &\rightarrow \mathbb{R} \\ (\tau, i, j) &\mapsto f(t_0 + \tau \Delta t, x_0 + i \Delta x, y_0 + j \Delta y) \end{aligned}$$

Repräsentiert man (wie üblich) reelle Zahlen durch genäherte Darstellung mit endlich vielen Bits, so kann man für ein festes t die Werte $g(t, i, j)$ als die Zustände der Zellen (i, j) in einer Konfiguration eines zweidimensionalen Zellularautomaten auffassen. Die Aufgabe besteht dann darin, für ihn eine Überföhrungsfunktion so zu definieren, dass sich nach ihrer einmaligen Anwendung in Zelle (i, j) der Wert $g(t + 1, i, j)$ – jedenfalls näherungsweise – ergibt. Aus mehreren Gründen ist diese Aufgabe im allgemeinen nicht exakt lösbar.

- Für die wenigsten Funktionen ist der Funktionswert an einer Stelle zum Zeitpunkt $t + \Delta t$ aus Funktionswerten zum Zeitpunkt t an einigen benachbarten Stellen exakt bestimmbar. In einigen Fällen gelingt aber immerhin eine befriedigende Näherung.
- Im allgemeinen kommt es durch die Darstellung reeller Zahlen mit endlich vielen Bits zu Ungenauigkeiten. Auf diesen Aspekt werden wir am Ende des Kapitels ganz kurz eingehen. Für die allgemeine Fragestellung der *Stabilität* sei auf Fletcher (1991, Kapitel 4) verwiesen.

Im folgenden stellen wir zunächst einige allgemeine Überlegungen zur Berechnung von $f(t + \Delta t, x, y)$ aus $f(t, x, y)$ und „benachbarten“ Werten wie zum Beispiel $f(t, x + \Delta x, y + \Delta y)$ an. Das ganze wird auf etwas hinauslaufen, das man als *Finite-Differenzen-Verfahren* bezeichnet.

12.8 RECHNUNG. Für jede Variable $q \in \{t, x, y\}$, das entsprechende Δq und den Operator $Q \in \{T, X, Y\}$ liefert die Taylorentwicklung:

$$Qf = f + \Delta q \frac{\partial f}{\partial q} + \frac{(\Delta q)^2}{2} \frac{\partial^2 f}{\partial q^2} + \frac{(\Delta q)^3}{3!} \frac{\partial^3 f}{\partial q^3} + \dots$$

$$Q^-f = f - \Delta q \frac{\partial f}{\partial q} + \frac{(\Delta q)^2}{2} \frac{\partial^2 f}{\partial q^2} - \frac{(\Delta q)^3}{3!} \frac{\partial^3 f}{\partial q^3} \pm \dots$$

Mit dem Hintergedanken, dass Δq immer „sehr klein“ ist (also „gegen Null geht“, so dass höhere Potenzen gegenüber niedrigeren vernachlässigbar sind), schreiben wir dies wie folgt:

$$Qf = f + \Delta q \frac{\partial f}{\partial q} + \frac{(\Delta q)^2}{2} \frac{\partial^2 f}{\partial q^2} + \frac{(\Delta q)^3}{3!} \frac{\partial^3 f}{\partial q^3} + O((\Delta q)^4) \quad (12.2)$$

$$\frac{\partial f}{\partial q} \approx \frac{Q - I}{\Delta q} f \quad (\text{Fehler: } O(\Delta q)) \quad (12.3)$$

$$Q^-f = f - \Delta q \frac{\partial f}{\partial q} + \frac{(\Delta q)^2}{2} \frac{\partial^2 f}{\partial q^2} - \frac{(\Delta q)^3}{3!} \frac{\partial^3 f}{\partial q^3} + O((\Delta q)^4) \quad (12.4)$$

$$Qf - Q^-f = 2\Delta q \frac{\partial f}{\partial q} + O((\Delta q)^3) \quad (12.5)$$

$$\frac{\partial f}{\partial q} \approx \frac{Q - Q^-}{2\Delta q} f \quad (\text{Fehler: } O((\Delta q)^2)) \quad (12.6)$$

Für die zweite Ableitung ergibt sich:

$$Qf + Q^-f = 2f + (\Delta q)^2 \frac{\partial^2 f}{\partial q^2} + O((\Delta q)^4)$$

$$\frac{\partial^2 f}{\partial q^2} \approx \frac{Q - 2I + Q^-}{(\Delta q)^2} f \quad (\text{Fehler: } O((\Delta q)^2)) \quad (12.7)$$

12.9 BEISPIEL.

$$\frac{\partial^2 f}{\partial x^2} \approx \frac{(f(t, x + \Delta x, y) - f(t, x, y)) - (f(t, x, y) - f(t, x - \Delta x, y))}{(\Delta x)^2} \quad (12.8)$$

12.10 BEISPIEL. (DIFFUSION) Es sei f eine Funktion mit $\dot{f} = D\nabla^2 f$, also

$$\frac{\partial f}{\partial t} = D \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right)$$

für eine Konstante D . Hierbei handelt es sich um die partielle Differentialgleichung, die *Diffusion* beschreibt. Nach obiger Rechnung ergibt sich dann unter Benutzung von 1.3 und 1.7:

$$\frac{T-I}{\Delta t} f \approx D \frac{X-2I+X^-}{(\Delta x)^2} f + D \frac{Y-2I+Y^-}{(\Delta y)^2} f,$$

also

$$\begin{aligned} f(t+\Delta t, x, y) &\approx f(t, x, y) \\ &+ D\Delta t \frac{(f(t, x+\Delta x, y) - f(t, x, y)) - (f(t, x, y) - f(t, x-\Delta x, y))}{(\Delta x)^2} \\ &+ D\Delta t \frac{(f(t, x, y+\Delta y) - f(t, x, y)) - (f(t, x, y) - f(t, x, y-\Delta y))}{(\Delta y)^2} \end{aligned}$$

Eine weniger fehlerträchtige Approximation ergibt sich bei Verwendung von 1.6 statt 1.3. Man erhält dann

$$\frac{T-T^-}{2\Delta t} f \approx D \frac{X-2I+X^-}{(\Delta x)^2} f + D \frac{Y-2I+Y^-}{(\Delta y)^2} f,$$

also

$$\begin{aligned} f(t+\Delta t, x, y) &\approx f(t-\Delta t, x, y) \\ &+ 2D\Delta t \frac{(f(t, x+\Delta x, y) - f(t, x, y)) - (f(t, x, y) - f(t, x-\Delta x, y))}{(\Delta x)^2} \\ &+ 2D\Delta t \frac{(f(t, x, y+\Delta y) - f(t, x, y)) - (f(t, x, y) - f(t, x, y-\Delta y))}{(\Delta y)^2} \end{aligned}$$

Wie man sieht, zahlt man für die bessere Genauigkeit mit Speicherplatz: Neben den Funktionswerten zum Zeitpunkt t wie auch schon im vorangegangenen Fall benötigt man zusätzlich noch den „alten“ Funktionswert $f(t-\Delta t, x, y)$.

12.11 RECHNUNG. Eine andere Möglichkeit, Diffusion anzunähern, ist Mittelwertbildung in der Nachbarschaft jeder Zelle. Im folgenden sei $N = M_r^{(2)}$. Insbesondere ist also die Nachbarschaft symmetrisch, d.h. für jedes $\mathbf{n} \in N$ ist auch $-\mathbf{n} \in N$. Der Einfachheit halber nehmen wir im folgenden an, dass $\Delta x = \Delta y$. Wir betrachten eine Funktion $f(\mathbf{x})$ und

$$\bar{f}(\mathbf{x}) := \frac{1}{|N|} \sum_{\mathbf{n} \in N} f(\mathbf{x} + \Delta x \mathbf{n}).$$

Taylorentwicklung für die $f(\mathbf{x} + \Delta x \mathbf{n})$ liefert

$$\begin{aligned} \bar{f}(\mathbf{x}) &= \frac{1}{|N|} \sum_{\mathbf{n} \in N} \sum_{k=0}^{\infty} \frac{1}{k!} \left((\Delta x \mathbf{n} \cdot \nabla)^k f \right) (\mathbf{x}) \\ &= \sum_{k=0}^{\infty} \frac{1}{|N|} \frac{1}{k!} \sum_{\mathbf{n} \in N} \left((\Delta x \mathbf{n} \cdot \nabla)^k f \right) (\mathbf{x}) \end{aligned}$$

Wegen der Symmetrie der Nachbarschaft ist die hintere Summe für alle ungeraden k gleich 0 und man erhält näherungsweise

$$\bar{f}(\mathbf{x}) \approx f(\mathbf{x}) + \frac{1}{2|\mathbf{N}|} \sum_{\mathbf{n} \in \mathbf{N}} \left((\Delta x \mathbf{n} \cdot \nabla)^2 f \right) (\mathbf{x}) = f(\mathbf{x}) + \frac{(\Delta x)^2}{2|\mathbf{N}|} \sum_{\mathbf{n} \in \mathbf{N}} \left((\mathbf{n} \cdot \nabla)^2 f \right) (\mathbf{x}) \quad (12.9)$$

Nach Voraussetzung ist für jedes $\mathbf{n} = (i, j) \in \mathbf{N}$ auch $\mathbf{n}' = (i, -j) \in \mathbf{N}$. Für $j \neq 0$ liefern die beiden zugehörigen Summanden zusammen einen Beitrag von

$$\begin{aligned} (\mathbf{n} \cdot \nabla)^2 + (\mathbf{n}' \cdot \nabla)^2 &= i^2 \frac{\partial^2}{\partial x^2} + 2ij \frac{\partial}{\partial x} \frac{\partial}{\partial y} + j^2 \frac{\partial^2}{\partial y^2} + i^2 \frac{\partial^2}{\partial x^2} - 2ij \frac{\partial}{\partial x} \frac{\partial}{\partial y} + j^2 \frac{\partial^2}{\partial y^2} \\ &= 2i^2 \frac{\partial^2}{\partial x^2} + 2j^2 \frac{\partial^2}{\partial y^2} \end{aligned}$$

zu der Summe. Die gemischten Glieder $\frac{\partial}{\partial x} \frac{\partial}{\partial y}$ fallen also alle weg. Daher ergibt sich

$$\begin{aligned} \sum_{\mathbf{n} \in \mathbf{N}} (\mathbf{n} \cdot \nabla)^2 &= 2 \left(\sum_{i=1}^r \sum_{j=1}^r 2i^2 \frac{\partial^2}{\partial x^2} + 2j^2 \frac{\partial^2}{\partial y^2} \right) + 2 \left(\sum_{i=1}^r i^2 \frac{\partial^2}{\partial x^2} \right) + 2 \left(\sum_{j=1}^r j^2 \frac{\partial^2}{\partial y^2} \right) \\ &= 4 \left(r \sum_{i=1}^r i^2 \frac{\partial^2}{\partial x^2} + r \sum_{j=1}^r j^2 \frac{\partial^2}{\partial y^2} \right) + \frac{2r(r+1)(2r+1)}{6} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \\ &= \frac{(4r+2)r(r+1)(2r+1)}{6} \nabla^2. \end{aligned}$$

Somit erhält man aus 1.9:

$$\begin{aligned} \bar{f}(\mathbf{x}) &\approx f(\mathbf{x}) + \frac{(\Delta x)^2}{2(2r+1)^2} \frac{2(2r+1)^2 r(r+1)}{6} (\nabla^2 f) (\mathbf{x}) \\ &= f(\mathbf{x}) + (\Delta x)^2 \frac{r(r+1)}{6} (\nabla^2 f) (\mathbf{x}) \end{aligned}$$

beziehungsweise

$$(\nabla^2 f) (\mathbf{x}) \approx \frac{6}{(\Delta x)^2 r(r+1)} (\bar{f}(\mathbf{x}) - f(\mathbf{x}))$$

und es folgt:

$$Tf \approx f + \Delta t \frac{\partial f}{\partial t} = f + \Delta t D \nabla^2 f \approx \left(1 - \frac{6D\Delta t}{(\Delta x)^2 r(r+1)} \right) f + \frac{6D\Delta t}{(\Delta x)^2 r(r+1)} \bar{f} \quad (12.10)$$

Insbesondere gilt also für $\frac{6D\Delta t}{(\Delta x)^2 r(r+1)} = 1$: $Tf \approx \bar{f}$.

Oder andersherum: Verwendet man Mittelwertbildung als Überföhrungsfunktion, dann simuliert man näherungsweise Diffusion mit dem Diffusionskoeffizienten $D = \frac{(\Delta x)^2 r(r+1)}{\Delta t \cdot 6}$.

Um verschiedene Diffusionskoeffizienten zu simulieren, kann man also zum Beispiel die GröÖe der Nachbarschaft (i. e. r) variieren. Eine andere Möglichkeit besteht darin, statt der einfachen Durchschnittsbildung eine gewichtete Summe der Werte in der Nachbarschaft zu berechnen, also etwas von der Form $\bar{f}(\mathbf{x}) = \sum_{\mathbf{n} \in \mathbf{N}} a_{\mathbf{n}} f(\mathbf{x} + \Delta x \mathbf{n})$. Die Koeffizienten $a_{\mathbf{n}}$ darf man dabei natürlich nicht willkürlich wählen. Wirft man noch einmal einen Blick auf Gleichung 1.9 und berücksichtigt man, dass man möchte, dass $\sum_{\mathbf{x} \in \mathbf{R}} f(\mathbf{x})$ eine Invariante ist, so sieht man relativ leicht ein, dass es vernünftig ist zu verlangen, dass $\sum_{\mathbf{n} \in \mathbf{N}} a_{\mathbf{n}} \mathbf{n} = \mathbf{0}$ und $\sum_{\mathbf{n} \in \mathbf{N}} a_{\mathbf{n}} = 1$. Je nachdem welche zusätzlichen Eigenschaften die $a_{\mathbf{n}}$ erfüllen, ist die Bestimmung des Diffusionskoeffizienten mehr oder weniger einfach. Eine genauere Diskussion findet sich zum Beispiel in der Diplomarbeit von Merkle 1997.

- 12.12 ÜBUNG. Versuchen Sie, für *eindimensionale* Zellularautomaten diese Rechnung durchzuführen. Nehmen Sie zusätzlich an, dass für alle \mathbf{n} gilt: $a_{\mathbf{n}} = a_{-\mathbf{n}}$.
- 12.13 ALGORITHMUS. Das Schöne an der eben angegebenen Näherungslösung ist, dass Mittelwerte (unter Umständen) leicht auszurechnen sind. Es sei $N = M_r$, f irgendeine Funktion und $g = |N|\bar{f}$. Betrachten wir zunächst den eindimensionalen Fall. Dann ist:

$$\begin{aligned}
 g(x) &= f(x-r) + f(x-r+1) + \cdots + f(x) + \cdots + f(x+r-1) + f(x+r) \\
 &= -f(x-r-1) + f(x-r-1) \\
 &\quad + f(x-r) + f(x-r+1) + \cdots + f(x) + \cdots + f(x+r-1) + f(x+r) \\
 &= -f(x-r-1) \\
 &\quad + f(x-1-r) + f(x-r) + f(x-r+1) + \cdots + f(x) + \cdots + f(x-1+r) \\
 &\quad + f(x+r) \\
 &= -f(x-r-1) + g(x-1) + f(x+r).
 \end{aligned}$$

Wenn man also z. B. in einem Simulationsprogramm die $g(x)$ *sequentiell*, also nacheinander für aufsteigende x berechnet, dann genügen jeweils zwei Additionen (statt $2r$ Additionen).

Im Zweidimensionalen kann man etwas Entsprechendes machen. Setzt man

$$\begin{aligned}
 h(x, y) &= \sum_{j=-r}^r f(x, y+j) \\
 g(x, y) &= \sum_{i=-r}^r h(x+i, y),
 \end{aligned}$$

dann ist $g = |N|\bar{f}$. Auch hier lassen sich in Anlehnung an die Idee für den eindimensionalen Fall sowohl die $h(x, y)$ als auch die $g(x, y)$ effizient, nämlich mit insgesamt nur 4 statt $(2r+1)^2 - 1$ Additionen (bei der Hauruckmethode) pro Argumentpaar berechnen:

```

for (x=xmin; x<xmax; x++)
  for (y=ymin; y<ymax; y++)
  {
    h[x+r,y] = h[x+r,y-1] - f[x+r,y-r-1] + f[x+r,y+r];
    g[x,y]   = g[x-1,y]   - h[x-r-1,y]   + h[x+r,y];
  }

```

Dabei ist es offensichtlich notwendig, die Felder auf geeignete Weise zu dimensionieren und zu initialisieren!

- 12.14 BEISPIEL. (WELLENAUSBREITUNG) Es sei f eine Funktion mit $\ddot{f} = c^2 \nabla^2 f$, also

$$\frac{\partial^2 f}{\partial t^2} = c^2 \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right)$$

für eine Konstante c . Hierbei handelt es sich um die partielle Differentialgleichung, die *Wellenausbreitung* beschreibt. Zum Beispiel ist im Falle von Schall f der Schalldruck und c die Schallgeschwindigkeit.

Für die Berechnung von $c^2 \nabla^2 f$ benutzt man eine der schon behandelten Näherungen. Es stellt sich die Frage, wie man von $\frac{\partial^2 f}{\partial t^2}$ zu einem Näherungswert für Tf kommt. Eine Möglichkeit besteht in der Benutzung von Formel 1.7, die

$$Tf \approx 2f - T^-f + (\Delta t)^2 \frac{\partial^2 f}{\partial t^2}$$

liefert. Wie man sieht, muss man dann in jeder Zelle neben $f(t, x, y)$ auch noch $f(t - \Delta t, x, y)$ speichern.

Statt dessen kann man aber auch Formel 1.3 benutzen und eine Näherung für $g := \frac{\partial f}{\partial t}$ mitführen:

$$\begin{aligned} T f &\approx f + \Delta t \frac{\partial f}{\partial t} \approx f + \Delta t g \\ T g &\approx g + \Delta t \frac{\partial g}{\partial t} = g + \Delta t \frac{\partial^2 f}{\partial t^2} \end{aligned}$$

Oder man greift auf Formel 1.4 in der folgenden Form zurück:

$$\begin{aligned} T^- f &\approx f - \Delta t \frac{\partial f}{\partial t} \\ f &\approx T^- f + \Delta t g \\ T f &\approx f + \Delta t T g \end{aligned}$$

Damit ergibt sich

$$\begin{aligned} T g &\approx g + \Delta t \frac{\partial g}{\partial t} = g + \Delta t \frac{\partial^2 f}{\partial t^2} \\ T f &\approx f + \Delta t T g \end{aligned}$$

Diese letzte Version wurde von Magnussen 1993 in seiner Diplomarbeit verwendet.

12.15 ÜBUNG. Es ist sehr lehrreich, die verschiedenen Versionen auszuprobieren und miteinander zu vergleichen. Tun Sie das!

12.16 BEISPIEL. (BELOUSOV-ZHABOTINSKI-REAKTION) Die BZ-Reaktion ist ein Beispiel für eine chemische oszillierende Reaktion, die man auch als *chemische Uhr* bezeichnet. Sie ist bislang eine der ganz wenigen, die man praktisch vollständig versteht. Es sind 21 verschiedene Substanzen an insgesamt 18 Elementarreaktionen beteiligt.

In recht guter Näherung lassen sich die Abläufe durch den sogenannten „Oregonator“ (von Field, Körös und Noyes) beschreiben, in dem nur noch drei Konzentrationen $u(t)$, $v(t)$ und $w(t)$ eine Rolle spielen (siehe zum Beispiel Kapitel 7 in dem Buch von Murray 1993):

$$\begin{aligned} \frac{du}{dt} &= k_1 a v - k_2 u v + k_3 a u - k_4 u^2 \\ \frac{dv}{dt} &= -k_1 a v - k_2 u v + f k_5 w \\ \frac{dw}{dt} &= 2k_3 a u - k_5 w \end{aligned}$$

Dabei sind natürlich die Konstanten a, f, k_1, \dots, k_5 geeignet zu wählen.

Wenn man die BZ-Reaktion geeignet auf ein annähernd zweidimensionales Gebiet einschränkt, dann entstehen Muster wie zum Beispiel rotierende Spiralen. Für deren Modellierung müssen die Konzentrationen der beteiligten Substanzen dann offensichtlich auch ortsabhängig gemacht werden (siehe auch das folgende Kapitel.)

Eine ganze Klasse von partiellen Differentialgleichungssystemen, die zum Beispiel in der Biologie eine wichtige Rolle als Modelle für Musterbildungsprozesse spielen, sind die sogenannten **Reaktions-Diffusions-Systeme** (RDS). Einer der allerersten, grundlegenden Aufsätze zu diesem

Thema stammt von Turing 1952, der sich damit beschäftigt hat, wie wohl die regelmäßige Anordnung von Tentakeln beim Süßwasser-Polypen *Hydra* zu erklären sei. Es war dies die letzte wissenschaftliche Veröffentlichung von Turing. Im übrigen sei auf das umfangreiche Buch von Murray 1993 und das reich bebilderte von Meinhardt 1995 hingewiesen. Die beiden Autoren haben sich zum Beispiel mit der Modellierung der Entstehung von Mustern auf Säugetierfellen bzw. Muscheln beschäftigt.

- 12.17 BEISPIEL. (REAKTIONS-DIFFUSIONS-SYSTEME) Im allgemeinen haben Reaktions-Diffusions-Systeme die folgende Struktur, die auch den Namen plausibel macht:

$$\begin{aligned}\frac{\partial f_1}{\partial t} &= F_1(f_1, f_2, \dots, f_k) + D_1 \nabla^2 f_1 \\ \frac{\partial f_2}{\partial t} &= F_2(f_1, f_2, \dots, f_k) + D_2 \nabla^2 f_2 \\ &\vdots \\ \frac{\partial f_k}{\partial t} &= F_k(f_1, f_2, \dots, f_k) + D_k \nabla^2 f_k\end{aligned}$$

- 12.18 BEISPIEL. (RDS VON TURING)

$$\begin{aligned}F_1(f_1, f_2) &= f_1 f_2 - f_1 - 12 \\ F_2(f_1, f_2) &= -f_1 f_2 + 16\end{aligned}$$

Wie zu Beginn dieses Kapitels schon angedeutet, kann man gegen die auf den vergangenen Seiten beschriebene Vorgehensweise einwenden, dass die verwendeten Zustandsmengen *sehr* groß sind. Zwar kennt man Methoden, um jeden Zellularautomaten in einen äquivalenten umzuwandeln, bei dem jede Zelle nur noch zwei Zustände einnehmen kann, man bezahlt bei ihnen aber immer mit einer immensen Vergrößerung der Nachbarschaft.

Den Einwand noch besser auf den Punkt bringt daher die Formulierung, dass wir vorne in Kauf genommen haben, dass dort der Zustand jeder Zelle von vielen Nachbarbits abhängt (sei es nun von je vielen Bits in wenigen Zellen oder von je wenigen Bits in vielen Zellen).

Was kann man tun, um diese „Abhängigkeit“ zu verringern? Mit der Möglichkeit, in jeder Zelle nur Platz für einige wenige „Partikel“ (je 1 Bit) vorzusehen, die lokal miteinander interagieren, werden wir uns im nächsten Kapitel beschäftigen. Zum Abschluss dieses Kapitels wollen wir noch eine andere Vorgehensweise vorstellen.

- 12.19 (RUNDUNGSFEHLER) Grund für die vielen Bits ist der Wunsch, möglichst genau rechnen zu können. Würde man zum Beispiel nur mit 6-Bit-Festkommazahlen, also o. B. d. A. etwa Ganzzahlen, rechnen, dann träten bei naiver Herangehensweise zu viele Rundungsfehler auf, die schnell dazu führen, dass die Simulationsergebnisse nichts mehr mit der Realität zu tun hätten.

Als Beispiel betrachte man die folgende Entwicklung eines eindimensionalen Zellularautomaten, bei dem Durchschnittsbildung in der $H_1^{(1)}$ -Nachbarschaft mit Ganzzahlarithmetik berechnet wird:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 9 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 3 | 3 | 3 | 0 | 0 | 0 |
| 0 | 0 | 1 | 2 | 3 | 2 | 1 | 0 | 0 |
| 0 | 0 | 1 | 2 | 2 | 2 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 2 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Nach wenigen Schritten wird eine Konfiguration erreicht, in der – bedingt durch das ständige Abrunden – „alles verschwunden“ ist.

Die naheliegende Vorgehensweise, um diesen Effekt zu vermeiden, ist offensichtlich, nicht immer in die gleiche Richtung zu runden. So etwas wie „kaufmännisches Runden“ hilft in obigem Beispiel freilich nicht. Dann würde nämlich nach wenigen Schritten eine sich nicht weiter verändernde Konfiguration erreicht werden:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|--|
| 0 | 0 | 0 | 0 | 9 | 0 | 0 | 0 | 0 | |
| 0 | 0 | 0 | 3 | 3 | 3 | 0 | 0 | 0 | |
| 0 | 0 | 1 | 2 | 3 | 2 | 1 | 0 | 0 | |
| 0 | 0 | 1 | 2 | 2 | 2 | 1 | 0 | 0 | |
| 0 | 0 | 1 | 2 | 2 | 2 | 1 | 0 | 0 | |

Weimar und Boon 1994 haben beobachtet, dass man bei Zellularautomaten – jedenfalls in manchen Fällen – gute Ergebnisse erzielen kann, wenn man die Rundung *zufällig* durchführt, aber nicht völlig willkürlich.

12.20 DEFINITION Ein *probabilistischer Zellularautomat*, oder auch *stochastischer Zellularautomat*, ist dadurch gekennzeichnet, dass die lokale Überföhrungsfunktion von der Form

$$\delta : Q^N \rightarrow [0; 1]^Q$$

ist und für alle $l \in Q^N$ gilt: $\sum_{q \in Q} \delta(l)(q) = 1$. Bei einem Schritt wählt eine Zelle mit lokaler Konfiguration l mit Wahrscheinlichkeit $\delta(l)(q)$ den Zustand q als Nachfolgezustand. \diamond

12.21 Zwei Bemerkungen erscheinen angebracht:

1. Man beachte, dass verschiedene Zellen, die in einer Konfiguration die *gleiche* lokale Konfiguration in ihrer Nachbarschaft beobachten, in einem Schritt eines probabilistischen Zellularautomaten in *verschiedene* Nachfolgezustände übergehen können.
2. Intuitiv ist eine probabilistische Überföhrungsfunktion etwas komplizierteres als eine deterministische Überföhrungsfunktion: „Wer würfelt?“ Je nachdem, wie die Wahrscheinlichkeiten gewählt sind, kann man das aber unter günstigen Umständen auch mit ein oder zwei zusätzlichen Bits je Zelle mit Hilfe einer deterministischen Überföhrungsfunktion realisieren (siehe nächstes Kapitel).

12.22 Im oben angesprochenen Fall, dass mit Hilfe eines probabilistischen Zellularautomaten Rundungsprobleme umgehen möchte, kann man nun wie folgt vorgehen. Nehmen wir an, man müsse ein Rechenergebnis $k + \varepsilon$ mit $k \in \mathbb{N}$ und $\varepsilon \in [0; 1[$ auf eine ganze Zahl „runden“. Dann lege man zu Beginn Zahlen k_i und zugehörige Wahrscheinlichkeiten $p_i(\varepsilon)$ fest, so dass $\sum_i p_i(\varepsilon) = 1$ ist und außerdem $\sum_i k_i p_i(\varepsilon) = k + \varepsilon$, und wähle mit Wahrscheinlichkeit $p_i(\varepsilon)$ den Wert k_i .

Die einfachste solche Methode besteht darin, stets mit Wahrscheinlichkeit ε auf $k + 1$ zu runden und mit Wahrscheinlichkeit $1 - \varepsilon$ auf k . Der Erwartungswert für die gewählte Zahl ist dann nämlich: $\varepsilon(k + 1) + (1 - \varepsilon)k = k + \varepsilon$. Man macht also *sehr salopp* formuliert sozusagen „im Mittel keinen Rundungsfehler“.

Zusammenfassung

- Geeignete Diskretisierung partieller Differentialgleichungen liefert manchmal Zellularautomaten, die gute Näherungen sind.

- Rundungsfehler kann man manchmal durch probabilistisches Runden (in entsprechenden Zellularautomaten) verkleinern.

Literatur

- Fletcher, C. A. J. (1991). *Computational Techniques for Fluid Dynamics*. 2nd. Bd. 1. Springer-Verlag.
- Magnussen, B. (1993). „Wellen in Zellularautomaten“. Diploma thesis. Universität Karlsruhe.
- Meinhardt, H. (1995). *The Algorithmic Beauty of Sea Shells*. Berlin: Springer.
- Merkle, Daniel (1997). „Theoretische und praktische Untersuchungen an stochastischen Zellularautomaten“. Diploma thesis. Universität Karlsruhe.
- Murray, J. D. (1993). *Mathematical Biology*. 2nd. Bd. 19. Biomathematics Texts. Springer-Verlag.
- Turing, A. M. (1952). „The chemical basis of morphogenesis“. In: *Phil. Trans. Roy. Soc. Lond.* B237, S. 37–72.
- Weimar, Jörg Richard und Jean-Pierre Boon (1994). „Class of Cellular Automata for Reaction-Diffusion Systems“. In: *Physical Review E* 49.2, S. 1749–1752.