Kapitel 12

Partielle Differentialgleichungen

Fast jedes physikalische Problem, bei dem Systemgrößen von Ort und Zeit abhängen, läßt sich in Form einer (oder eines Systems von) partiellen Differentialgleichung(en) darstellen. Beispiele dafür sind die schon bekannte Diffusionsgleichung, die elektromagnetischen Wellengleichungen, hydrodynamische Gleichungen, und die Schrödinger Gleichung aus der Quantenmechanik. Bis auf die einfachsten Spezialfälle sind solche Problemstellungnen nicht analytisch zu lösen, man ist also (wieder) auf numerische Hilfsmittel angewiesen. Man diskretisiert die partiellen Differentialgleichungen im Ort und in der Zeit und erhält ein – unter Umständen riesiges – System von Differenzengleichgungen, wie in diesem Kapitel an einigen Beispielen gezeigt wird.

Eine Vielzahl der physikalisch relevanten partiellen Differentialgleichungen enhthalt nur Ableitungen bis zur zweiten Ordnung, und kann deshalb in folgendes grobe Typenschema klassifiziert werden.

- *Parabolische Differentialgleichungen* enthalten eine erste Ableitung einer Variablen und zweite Ableitungen der Anderen. Beispiele sind die Diffusionsgleichung und die zeitabhängige Schrödingergleichgung, die beide eine erste Zeit- und zweite Ortsableitung enthalten.

- *Elliptische Differentialgleichungen* enthalten zweite Ableitungen aller Variablen, wobei diese dasselbe Vorzeichen haben wenn man sie auf eine Seite bringt. Beispiele hierfür sind die Poisson Gleichung für ein elektrostatisches Potential und die zeitunabhängige Schrödingergleichung, beide mit zwei oder mehr Ortsvariablen. Beachtenswert ist, daß der stationäre Zustand einer parabolischen Differentialgleichung eine elliptische Differentialgleichung darstellt.

- *Hyperbolische Differentialgleichungen* enthalten zweite Ableitungen aller Variablen mit verschiedenen Vorzeichen. Ein Beispiel ist die Wellengleichung einer gespannten Saite. Die Lösungsverfahren hyperbolischer Differentialgleichungen beinhalten oft eine Eigenwert und Eigenmodenanalyse, da man oft weiß daß sich die Lösungen aus Schwingungen zusammensetzen.

12.1 Elliptische partielle Differentialgleichungen

Wegen der etwas einfacheren Form elliptischer Differentialgleichugen (DGL) und da man elliptische Differentialgleichungen erhält wenn man z.B. die Zeitableitung zu Null setzt, also einen zeitlich unveränderlichen Zustand fordert, werden hier zuerst elliptische DGL behandelt.

Insbesondere sind wir an der Lösung der Gleichung

$$-\left[\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right]\phi = S(x,y) \tag{12.1}$$

interessiert, wozu allerdings noch die Spezifikation von Randbedingungen nötig ist. In einem Gleichgewichts-Wärmeleitungsproblem (Diffusion) entspricht das Feld ϕ der Temperatur und S der lokalen Wärmeerzeugungs- bzw. -verlustrate. Die Randbedingung soll hier in *Dirichlet-Form* angegeben werden, d.h. ϕ wird am Rand eines geschlossenen Gebietes fixiert (und evtl. zusätzlich an geschlossenen Kurven innerhalb des Gebiets). Eine andere mögliche Randbedingung hätte die *Neumann-Form*, d.h. die Ableitung von ϕ wird am Rand vorgegeben.

12.1.1 Diskretisierung

Im folgenden soll das quadratische Volumen der Länge $L = N$ mit einem kartesischen Quadratgitter von $i, j = 0, ..., N$ diskretisiert werden. Mit anderen Worten wird das System in $N \times N$ gleichgroße Quadrate der Seitenlänge h unterteilt, bzw. man hat $N+1 \times N+1$ Gitterpunkte. Nun definieren wir $\phi_{ij} = \phi(x_i, y_j)$ und $S_{ij} = S(x_i, y_j)$, mit $x_i = ih$ und $y_i = jh$, und verwenden die Näherung für die zweite Ableitung aus Kapitel 6.1.2, woraus sich ergibt:

$$-\left[\frac{\phi_{i+1\,j}-2\phi_{ij}+\phi_{i-1\,j}}{h^2}+\frac{\phi_{i\,j+1}-2\phi_{ij}+\phi_{i\,j-1}}{h^2}\right] = S_{ij}\ . \tag{12.2}$$

Verwendet man die Abkürzung

$$\partial_i^2\phi_{ij} = \phi_{i+1\,j} - 2\phi_{ij} + \phi_{i-1\,j} \tag{12.3}$$

und die analoge Abkürzung für $\partial_j^2\phi_{ij}$, so ergibt sich aus Gleichung 12.2 die kürzere Form:

$$-\left[\partial_i^2\phi_{ij} + \partial_i^2\phi_{ij}\right] = h^2 S_{ij}\ . \tag{12.4}$$

Diese Diskretisierung ist relativ einfach und intuitiv, aber manchmal leider zu ungenau. Erfordert das Problem höhere Genauigkeit oder mehr Flexibilität bei der Wahl des Gitters, so kann man andere Diskretisierungen mit einem Variationsverfahren ableiten [13.1].

12.1.2 Randbedingungen

In der Regel werden die Randbedingungen nicht genau auf den Gitterpunkten bekannt sein, d.h. die glatten Ränder eines realen Systems werden durch das Diskretisierungsgitter nur ungenau angenähert. Dieses Problem kann durch eine Gitterverfeinerung in der Nähe der Ränder oder durch Ableitungsdiskretisierung höherer Ordnung umgangen werden [13.1]. Der Einfachheit halber benutzen wir einfach ein quadratisches System, dessen Ränder genau mit den Randpunkten unseres Gitters übereinstimmen. Alle Einträge ϕ_{i0}, ϕ_{iN}, ϕ_{0j} und ϕ_{Nj} sind also vorgegeben. Für einen Punkt, z.B. $(N-1, j)$ mit $1 < j < N-1$, in der Nähe des rechten Rands ergibt sich aus Gleichung 12.4:

$$-\left[-4\phi_{N-1\,j} + \phi_{N-2\,j} + \phi_{N-1\,j+1} + \phi_{N-1\,j-1}\right] = h^2 S_{N-1\,j} + \phi_{N\,j} \ , \tag{12.5}$$

mit allen bekannten Größen auf der rechten Seite. Sind die Randpunkte festgelegt, ergibt sich somit ein lineares Gleichungssystem der Dimension $(N-1)^2$ in den Unbekannten ϕ

$$\mathbf{T} \cdot \mathbf{x} = \mathbf{s} \ , \tag{12.6}$$

welches allerdings für zwei- oder dreidimensionale Systeme sehr schnell unhandlich groß werden kann.

12.1.3 Iterative Lösungsverfahren

Um ein iteratives Lösungsverfahren zu begründen führen wir die Größe

$$E = \int_0^1 dx \int_0^1 dy \ \left[\frac{1}{2}(\nabla\phi)^2 - S\phi\right] \tag{12.7}$$

ein, die in der Elektrostatik den Sinn der Gesamtenergie des Systems hat ($\nabla\phi$ ist der Feldgradient und S ist die Ladungsdichte). Im Falle der Diffusionsgleichung gibt es keine gleichermassen angenehme Analogie, man kann E nur als eine nützliche Größe ansehen. Die Variation von E durch eine kleine Änderung von ϕ,

$$\delta E = \int_0^1 dx \int_0^1 dy \ \left[\nabla\phi \cdot \nabla\delta\phi - S\delta\phi\right] \tag{12.8}$$

kann partiell integriert werden. Da das dabei entstehende Linienintegral am Rand des Volumens verschwindet (solange die Randbedingung erfüllt ist) bleibt im Integral ein Term übrig, der genau Gleichung 12.1 entspricht. Die Forderung, daß E unverändert bleibt ist damit äquivalent zu der Aussage daß ϕ eine Lösung von Gleichung 12.2 ist. Mit anderen Worten ist E minimal wenn man die "richtige Lösung" ϕ einsetzt. Gleichung 12.7 läßt sich einfach diskretisieren, indem man für den Gradienten $\nabla\phi$ die Trapezregel (oder eine bessere Näherung) einsetzt

$$E = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[(\phi_{ij} - \phi_{i-1\,j})^2 + (\phi_{ij} - \phi_{i\,j-1})^2\right] - h^2 \sum_{i=1}^{N-1}\sum_{j=1}^{N-1} S_{ij}\phi_{ij} \tag{12.9}$$

Bei Verwendung der Trapezregel erhält man so die Gleichungen 12.4 wenn man fordert

$$\frac{\partial E}{\partial \phi_{ij}} = 0 \tag{12.10}$$

Aufgelöst nach den ϕ_{ij} ergibt sich

$$\phi_{ij} = \frac{1}{4}\left[\phi_{i+1\,j} + \phi_{i-1\,j} + \phi_{i\,j+1} + \phi_{i\,j-1} + h^2 S_{ij}\right] \;, \tag{12.11}$$

was noch keine Lösung von ϕ ergibt, aber stets zu einer Verbesserung führt wenn man sie auf alle ϕ_{ij} nacheinander (z.B. von links nach rechts und von oben nach unten) anwendet. Man beachte, daß bei der Berechnung des nächsten Werts der vorherige überschrieben wird; das Feld muß also nicht zwischengespeichert werden. Die verallgemeinerte Form

$$\phi_{ij} = (1-w)\phi_{ij} + w\frac{1}{4}\left[\phi_{i+1\,j} + \phi_{i-1\,j} + \phi_{i\,j+1} + \phi_{i\,j-1} + h^2 S_{ij}\right] \;, \tag{12.12}$$

stellt die sog. Gauss-Seidel Iteration mit dem Gewicht w dar. Benutzt man Glg. 12.9 um die Änderung der Energie bei einem Schritt zu berechnen (und beachtet dabei, daß sich nur ϕ_{ij} ändert) erhält man einen Wert der immer positiv ist falls $0 < \omega < 2$. Man nähert sich also mit jedem Schritt dem Minimum der Funktion E. Welche Werte von w optimal sind wird in der Übung getestet.

Eine Realisierung der Gauss-Seidel Iteration findet sich in nachfolgendem Programm.

```
#include<iostream>
#include<fstream>
using std::cout;
using std::cin;

static const int NL=40;
static double world[NL+1][NL+1];    // define the lattice
static double source[NL+1][NL+1];   // define the source

int main()
{
  int ISTEP=200;
  double w=1.5;
  double h=1.0/NL;

  for(int i=0; i<=NL; i++ ){
    for(int j=0; j<=NL; j++ ){
      world[i][j]=0.1;                 // initial values
      source[i][j]=0;                  // source field
    }
  }
```

```
  source[NL/2][NL/2]=2.0;              // source in the center
  source[NL/4][NL/4]=-1.0;             // source in the center
  source[3*NL/4][3*NL/4]=-1.0;         // source in the center

  for( int it=1; it<=ISTEP; it++ ){    // iteration loop
     double wsum=0;
     for(int i=1; i<NL; i++ ){
        for(int j=1; j<NL; j++ ){      // Gauss-Seidel loop
           world[i][j]=(1.0-w)*  world[i][j] +
                          w/4*(  world[i+1][j] + world[i-1][j]
                               + world[i][j+1] + world[i][j-1]
                               + h*h*source[i][j] );
           wsum+=world[i][j];
        }
     }
     cout << it << ' ' << wsum << '\n';

     std::ofstream outfile1("diff4.dat");
     for(int i=0; i<=NL; i++ ){
       for(int j=0; j<=NL; j++ ){
       outfile1 << world [i][j] << '\n';
       }
       outfile1 << '\n';
     }
     outfile1.close();
  }
  return 0;
}
```

Das Ergebnis des Programms ist in Abb. 12.1 dargestellt. Als Randbedinung wurde $\phi = 0.1$ am Rand gewählt und es wurden Quellen ($S(20,20) = 2$) bzw. Senken ($S(10,10) = S(30,30) = -1$) in dem ansonsten quellfreien Gitter der Größe $N = 40$ angebracht.

12.2 Parabolische partielle Differentialgleichungen

Erweitert man die Diffusionsgleichung aus dem letzten Kapitel um eine Zeitableitung, so erhält man eine parabolische DGL

$$\frac{\partial \phi}{\partial t} = \left[\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right] \phi + S(x,y) \ , \tag{12.13}$$

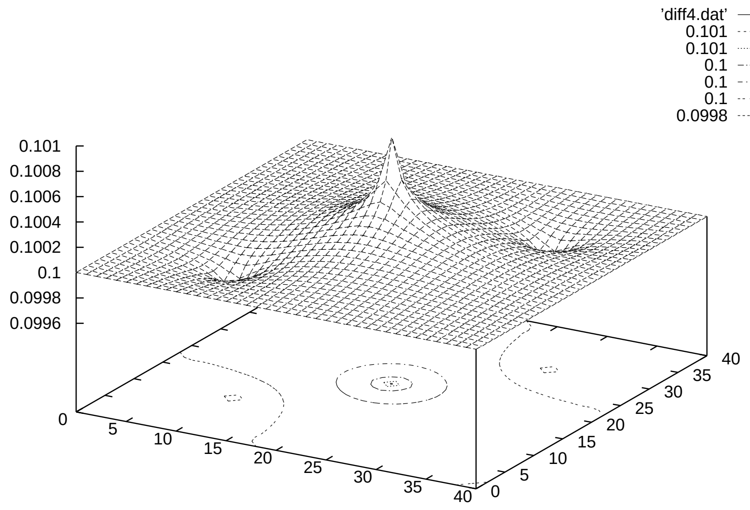

Abbildung 12.1: Lösung einer Diffusionsgleichung mit dem Gauss-Seidel Verfahren. Die Randbedinungen sind im Text angegeben.

die man in der einfachsten Form durch

$$\frac{\phi_{ij}(t+\Delta t) - \phi_{ij}(t)}{\Delta t} = \frac{1}{h^2}(\partial_i^2 + \partial_j^2)\phi_{ij}(t) + S_{ij} \tag{12.14}$$

diskretisiert. Aufgelöst nach den Werten von ϕ zur neuen Zeit $t + \Delta t$ hat man

$$\phi_{ij}(t+\Delta t) = \phi_{ij}(t) + \frac{\Delta t}{h^2}(-4\phi_{ij}(t) + \phi_{i-1\,j}(t) + \phi_{i+1\,j}(t) + \phi_{i\,j-1}(t) + \phi_{i\,j+1}(t)) + \Delta t S_{ij}\ . \tag{12.15}$$

Ein kleines Programm, mit dem man die Lösung der zweidimensionalen Diffusionsgleichung in Abwesehnheit von Senken ($S = 0$) erhält ist hier abgedruckt. Das Programm wurde aus dem Beispiel des letzten Abschnitts entwickelt und enthält nun zwei Felder, da die Berechnung des neuen Feldes diesmal das alte Feld nicht überschreiben darf.

```
#include<iostream>
#include<fstream>
using std::cout;
using std::cin;

static const int NL=40;
static double world[NL+1][NL+1];     // define the lattice
static double mirror[NL+1][NL+1];    // define the lattice image
static double source[NL+1][NL+1];    // define the source

int main()
```

```
{
  int ISTEP=100;
  double h=1.0/NL;
  double dt=0.0001;
  char cfile[20];

  for(int i=0; i<=NL; i++ ){
    for(int j=0; j<=NL; j++ ){
      world[i][j]=0.0;                // initial values
      source[i][j]=0.0;               // source field
    }
  }
  world[NL/2][NL/2]=NL*NL;

  for( int it=1; it<=ISTEP; it++ ){    // iteration loop
     double wsum=0;
     for(int i=1; i<NL; i++ ){
        for(int j=1; j<NL; j++ ){      // Time step ...
           mirror[i][j]=world[i][j] + dt/(h*h)*
                        (-4*world[i][j]+world[i-1][j]+world[i+1][j]+
                                        world[i][j-1]+world[i][j+1])
                        + dt*source[i][j];
           wsum+=mirror[i][j];
        }
     }
     for(int i=1; i<NL; i++ ){
        for(int j=1; j<NL; j++ ){      // Exchange old and new image
           world[i][j]=mirror[i][j];
        }
     }

     cout << it << ' ' << wsum << ' ';

     sprintf( cfile, "difft_%4.4d", it );
     cout << cfile << '\n';
     std::ofstream outfile1( cfile );
     for(int i=0; i<=NL; i++ ){
       for(int j=0; j<=NL; j++ ){
       outfile1 << world [i][j] << '\n';
       }
       outfile1 << '\n';
     }
     outfile1.close();
  }
```

```
  return 0;
}
```

In Abb. 12.2 ist die zeitabhängige Lösung für vier Zeitpunkte dargestellt. Diese Problem eignet sich gut zum Test des Programms, da die Summe aller ϕ_{ij} erhalten bleiben sollte. Wählt man den Zeitschritt zu groß (oder hat man einen Fehler einprogrammiert) so wird die Summe in der Regel nicht erhalten bleiben. In Abb. 12.2 stellt man fest, daß die Lösung (die bei $t = 0$ nur in der Mitte den Wert $\phi_{20\,20} = 2000$ hatte) mit der Zeit immer weiter auseinanderläuft und gleichzeitig flacher wird.

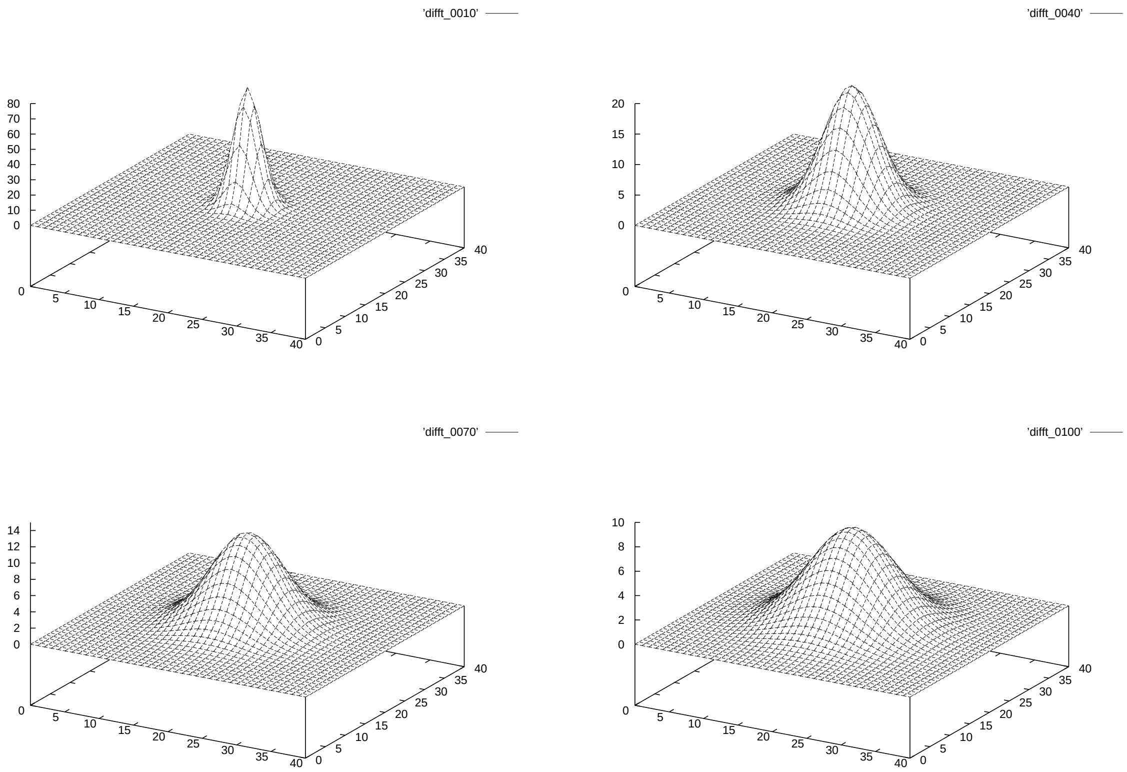

Abbildung 12.2: Zeitabhängige Lösung der Diffusionsgleichung nach 10, 40, 70 und 100 Schritten. Man beachte die verschiedenen vertikalen Achsenskalierungen.

Verwendet man die gleichen Quellen und Senken sowie die gleichen Randbedingungen $\phi_{\mathrm{Rand}} = 0.1$ wie im vorigen Abschnitt, erhält man nach ca. 10000 Zeitschritten genau dieselbe stationäre Lösung wie durch die Gauss-Seidel Iteration. Zusätzlich hat man nun allerdings auch die zeitliche Annäherung an das Gleichgewicht erhalten. Weiterhin sei bemerkt, daß diese Lösung auch durch direkte Monte-Carlo Simulation erhalten werden kann. ϕ ist dann die Wahrscheinlichkeit zur Zeit t ein Teilchen am Ort (x, y) zu finden.

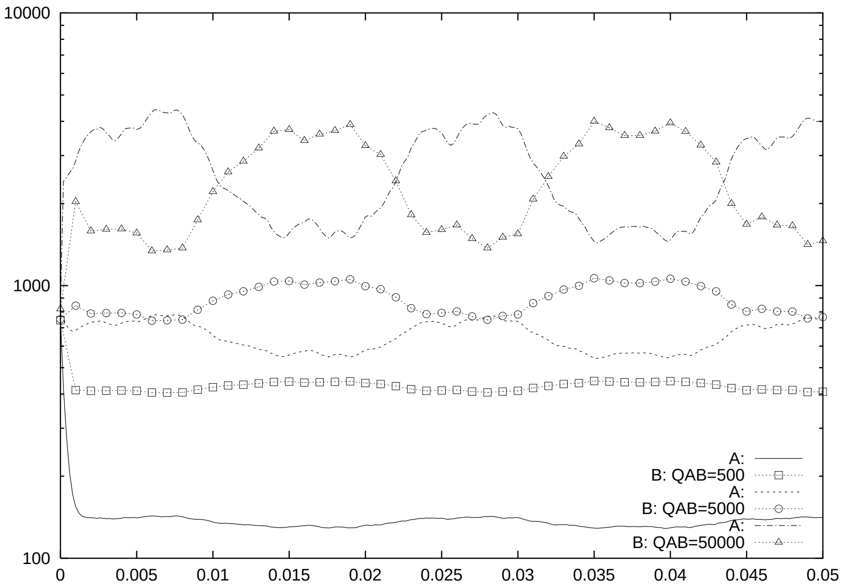

Abbildung 12.3: Lösung einer zeitabhängigen Diffusionsgleichung für zwei Spezies mit Erzeugungsrate Q_{AB} und Reaktionswahrscheinlichkeit $K_{AB} = 10^4$.

Das hier beschriebene Lösungsverfahren ist allerdings sehr einfach und ungenau. Bei komplizierteren Fragestellungen ist man auf genauere Verfahren angewiesen, die komplizierter zu implementieren sind. Trotzdem kann man mit diesem einfachen Verfahren bereits interesante physikalsiche Probleme bearbeiten. Bei einem vereinfachten Katalysator kann man annehmen, daß zwei Spezies A und B (z.B. Sauerstoff und Kohlenmonoxid) auf der Oberfläche diffundieren. Bei Kontakt reagieren A und B zu AB Molekülen (Kohlendioxid) und verschwinden vom System. Man löst dann die Diffusionsgleichung für beide Spezies und führt außerdem noch Zufuhr- und Reaktionsterme ein.

$$\frac{\partial \phi_A}{\partial t} = \left[\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right] \phi_A + Q_A(t) - K_{AB}\phi_A\phi_B \ , \tag{12.16}$$

$$\frac{\partial \phi_B}{\partial t} = \left[\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right] \phi_A + Q_B(t) - K_{AB}\phi_A\phi_B \ , \tag{12.17}$$

Die Felder ϕ_A und ϕ_B stehen dabei für die Dichte der einzelnen Spezies und der Quell-Term $Q(t)$ wird mit Zufallszahlen beschickt. Der Reaktionsterm hängt von der Dichte beider Spezies an einem Ort ab und hat die Form $K_{AB}\phi_A\phi_B$, wobei K_{AB} die Reaktionsrate ist. Beide Teilchenarten haben die gleichen Eigenschaften und existieren Anfangs zufällig verteilt auf dem Gitter. Die Lösung des resultierenden Gleichungssystems ist in Abb. 12.3 für verschiedene Erzeugungsraten Q_{AB} dargestellt. Man stellt Oszillationen des Systems fest, deren Periode mit zunehmender Zufuhrrate stark abnimmt. Obwohl die Zufuhr im Mittel für beide Spezies gleich ist kann eine Spezies über die andere siegen. Wenn z.B. von Spezies A viele Exemplare auf dem Gitter sind während nur wenige von Spezies B existieren, so werden all frisch zugeführten Teilchen von Spezies B sofort mit A reagieren und wieder verschwinden. Deshalb kann bei schwacher Zufuhr die Übermacht einer Spezies lange vorhalten. Diese Fragestellung ist Gegenstand aktueller Forschung (allerdings sind in realistischen Systemen viel mehr verschiedene Spezies vorhanden) und wird neben den in diesem Kapitel vorgestellten Methoden auch direkt mit Monte-Carlo Simulationen aus Kapitel 5 untersucht.

12.3 Hyperbolische partielle Differentialgleichungen

Eine hyperbolische Gleichung ist die Wellengleichung der lineare Kette des letzten Kapitels. Der Unterschied im Vorzeichen der zweiten Ableitungen führt zu komplexen Lösungen bei Ansetzen einer Exponentialfunktion, also zu Schwingungen [13.2]. Die Gleichung

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c^2 \frac{\partial^2 u(x,t)}{\partial x^2} \tag{12.18}$$

kann wie gehabt mit dem Zeitschritt Δt und der Gitterkonstante h diskretisiert werden zu

$$\frac{1}{\Delta t^2}\left[u(x,t+\Delta t) - 2u(x,t) + u(x,t-\Delta t)\right] = \frac{c^2}{h^2}\left[u(x+h,t) - 2u(x,t) + u(x-h,t)\right] \ . \tag{12.19}$$

Aufgelöst nach der gesuchten Größe $u(x, t + \Delta t)$ ergibt sich

$$u(x,t+\Delta t) = 2[1-b]u(x,t) + b[u(x+h,t) + u(x-h,t)] - u(x,t-\Delta t) \ , \tag{12.20}$$

wobei $b = (c\Delta t/h)^2$ dimensionslos ist. Diese Gleichung erinnert stark an das Verlet Integrationsschema aus Kapitel 3.2.1. Mit diesem Verfahren kann man nun die Dynamik einer Kette aneinandergekoppelter Massen untersuchen.

12.4 Weiterführende Literatur

Das Gebiet der partiellen Differentialgleichungen ist bei weitem zu umfangreich um im Rahmen dieser Vorlesung auch nur annähernd erfaßt werden zu können. Die hier vorgestellten Methoden stellen nur die Spitze des Eisbergs dar. Für tiefergehende Studien seien besonders die Bücher [13.1] und [13.2] empfohlen.

[13.1] S. E. Koonin and D. C. Meredith, *Computational Physics*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1990.
[13.2] H. Gould and J. Tobochnik, *An Introduction to Computer Simulation Methods*, Addison-Wesley Publishing Company, 1996.