Inhaltsverzeichnis

Wozu Simulationen?

Das klassische Instrument naturwissenschaftlicher Forschung ist das Experiment. Dies formuliert eine geeignete “Frage” an die Natur, und ein gut geplantes und durchgeführtes Experiment wird nach Messung und Datenauswertung eine Antwort auf diese Frage geben können. Die Schwierigkeiten experimenteller Wissenschaft besteht in der Regel in der Isolation von Störeinflüssen, die die Messungen und damit die Antwort beeinflussen und verfälschen können.

Experimentelles Naturverständnis wird begleitet von Theorie, in der wir unsere Vorstellungen der zentralen Vorgänge in dem Experiment in mathematischer Weise formulieren. Wir greifen dazu zunächst auf ein Modell zurück, dass alle wesentlichen physikalischen Einflussfaktoren erfassen soll und benutzen dann die Grundgleichungen der Physik — etwa die Schrödingergleichung oder die newtonschen Bewegungsgleichungen, um zu einer mathematischen Formulierung zu gelangen.

Nun kann man, und das ist die klassische theoretische Vorgehensweise, mit analytischen Methoden die sich ergebenden Gleichungen behandeln und ggf. durch Approximationen soweit vereinfachen, bis man eine “Lösung”, am besten in geschlossener Form, erhalten hat.

Simulationen geben dem Theoretiker ein Mittel in die Hand, das ihm erlaubt, diesen Prozess der sukzessiven Vereinfachung in einem frühen Stadium abzubrechen und das Problem mit dem Computer als Werkzeug noch auf einer “grundlegenden” Ebene, häufig sogar der Modellebene zu “simulieren”. Dabei geht natürlich einerseits Information über die analytische Abhängigkeit der Lösung von Problemparametern verloren, jedoch gewinnen wir andererseits dadurch, dass keine möglicherweise unkontrollierbaren Approximationen oder ungerechtfertigte Annahmen gemacht werden müssen.

Als Begriffsbestimmung halten wir fest:
Unter *(Computer-)Simulation* versteht man die typischerweise zeitlich und räumlich aufgelöste Nachbildung physikalischer Prozesse auf Digitalrechnern. Die dabei zum Einsatz kommenden Methoden bilden den Gegenstand dieser Vorlesung.

Simulationen

1. setzen auf “niedriger” Ebene, meist bei den Grundgleichungen des Systems an,

2. umgehen typische Approximationen, die in analytischen Herleitungen oft notwendig sind: Vereinfachung von Wechselwirkungstermen, Verwendung effekti-

ver Einteilchenpotentiale und Vernachlässigung von Mehrteilchenkorrelationen, möglicherweise unkontrollierbare “Annahmen” über das System, etc.,

3. bieten unmittelbaren Zugang zu Messgrößen,

4. können aber nur in dem Umfang Information liefern, wie dies die Grundgleichungen ermöglichen: keine Quanteneffekte bei Simulation der newtonschen Bewegungsgleichungen, möglicherweise völlig unzutreffende Resultate bei Start mit falschen Modellvorstellungen, etc.

Als Zielvorstellung steht hinter Simulationen das “Computerexperiment,” in dem möglichst alle Störeinflüsse eliminiert sind.

Der Übergang zu klassischen Methoden der numerischen Mathematik, wie sie etwa zur Lösung partieller Differentialgleichungen oder zur Lösung von Systemen gewöhnlicher Differentialgleichungen zur Anwendung kommen, ist fließend. Daher werden diese ebenfalls in begrenztem Umfang im Rahmen dieser Vorlesung behandelt.

Kapitel 1

Grundbegriffe der Statistik

Statistische Auswertemethoden werden immer gebraucht, wenn Messungen in Experimenten oder Computersimulationen gemacht werden und die Information über das System entweder nicht vollständig ist oder die Bearbeitung einer zu großen Menge an Daten verlangt. Diese “Unwissenheit” äußert sich in Variationen der Messgröße.

Als Beispiel betrachten wir ein stellvertretend für ein physikalisches System ein eindimensionales Verkehrsproblem, bei dem sich Autos auf einer einspurigen Straße mit unterschiedlichen Geschwindigkeiten fortbewegen:

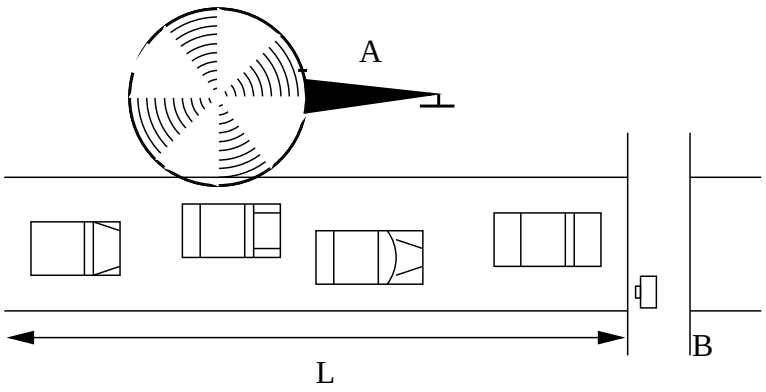


Abb. 1.1 *Geschwindigkeitsmessung an einer einspurigen Straße*

Wir können die Fahrzeuggeschwindigkeiten auf mehrere Arten bestimmen und zwar durch

1. ein Radarmeßgerät auf einer Brücke,

2. den Vergleich zweier kurz nacheinander gemachter Luftaufnahmen,

3. das Ablesen eines Tachometers oder Fahrtenschreibers in einem Fahrzeug

1.1 Verteilungen, Momente, empirische Schätzungen

Die Radarmessung liefert für jedes Auto i, das unter der Brücke hindurch fährt, einen Wert v_i der Geschwindigkeit.

1.1.1 Ensemblebegriff

Die gemessenen Werte für v_i ändern sich von Fahrzeug zu Fahrzeug, sind aber durch die "Verkehrssituation" charakterisiert. Bei Stau oder Ferienbeginn ergeben sich andere Werte als bei regulär fließendem Verkehr. Um die Vergleichbarkeit der Messungen zu gewährleisten, müssen wir daher die Rahmenbedingungen des Problems spezifizieren. Zur Untersuchung des Berufsverkehrs messen wir etwa täglich an Nicht-Feiertagen zwischen 07:00 und 09:00 Uhr, zur Bestimmung von Daten für die Gesamtabgasbelastung aber vielleicht über ein ganzes Jahr hinweg. Für physikalische Systeme geht man ebenso vor: wir betrachten etwa Systeme, die durch eine bestimmte Größe, Teilchenzahl, Energie oder auch nur durch die Probenpräparation charakterisiert sind.

Die Gesamtheit der Systeme, die diesen festgesetzten Rahmenbedingungen genügt, nennt man statistisches *Ensemble*.

1.1.2 Verteilung, Momente

Jedes Ensemble ist durch eine *Verteilungsdichte* der Werte v_i charakterisiert. Um diesen Begriff zu fassen, benutzen wir die Wahrscheinlichkeit $p(v \ldots v+\Delta v)$, einen Wert der Geschwindigkeit aus dem Intervall $I = [v, v+\Delta v]$ zu messen. Die Intervallbreite Δv ist dabei zunächst nicht festgelegt und auch nicht unbedingt klein. Die Verteilungsdichte $n(v)$ soll mit dieser Wahrscheinlichkeit über ein Integral zusammenhängen:

$$p(v \ldots v+\Delta v) = \int_v^{v+\Delta v} n(v')dv'. \tag{1.1}$$

Die Wahrscheinlichkeit $p(v \ldots v+\Delta v)$ ergibt sich als Grenzwert der relativen Häufigkeit,

$$p(v \ldots v+\Delta v) = \lim_{N\to\infty} \frac{N(v \ldots v+\Delta v)}{N} \tag{1.2}$$

eine Geschwindigkeit aus dem vorgegebenen Intervall zu messen. Hierbei steht N für die Gesamtzahl der gemessenen Geschwindigkeiten und $N(v \ldots v+\Delta v)$ für die Anzahl der Messergerbnisse in dem fraglichen Geschwindigkeitsintervall.

Wenn wir jetzt die Intervallbreite Δv so schmal wählen, dass sich $n(v)$ in dem Bereich nur unwesentlich ändert, dann können wir das Integral approximieren durch $n(v)\Delta v$ und erhalten

$$n(v)\Delta v = \lim_{N\to\infty} \frac{N(v \ldots v+\Delta v)}{N} \tag{1.3}$$

$$n(v) = \lim_{\Delta v\to 0} \lim_{N\to\infty} \frac{N(v \ldots v+\Delta v)}{N\Delta v} \tag{1.4}$$

In realen Situationen werden wir natürlich weder Δv infinitesimal klein noch N beliebig groß wählen können.

Die Verteilungsdichte genügt einer Normierungsbedingung, denn die Wahrscheinlichkeit, irgendeine Geschwindigkeit zu finden, ist immer 1 :

$$1 = \int_{-\infty}^{\infty} n(v)dv \tag{1.5}$$

Häufig reicht die Anzahl der verfügbaren Messwerte nicht aus, um $n(v)$ mit ausreichender Genauigkeit empirisch zu bestimmen. Darum wird $n(v)$ häufig durch ihre Momente charakterisiert. Es ist etwa der empirische Mittelwert der Geschwindigkeit

$$\overline{v} = \frac{1}{N}\sum_i v_i \tag{1.6}$$

und die Varianz

$$\overline{\Delta^2 v} = \frac{1}{N}\sum_i (v_i - \overline{v})^2. \tag{1.7}$$

Die Varianz ist der Mittelwert der quadratischen Abweichung der einzelnen Messwerte. Die Streuung oder (N-gewichtete) Standardabweichung ist die Wurzel daraus

$$\sigma = \sqrt{\overline{\Delta^2 v}}. \tag{1.8}$$

Alle diese Größen lassen sich mit Hilfe des Begriffes des p-ten Momentes M_p

$$M_p = \int_{-\infty}^{\infty} n(v)v^p dv \tag{1.9}$$

auf die Verteilungsdichte $n(v)$ zurückführen. Der theoretische Mittelwert einer Größe ist nicht anderes als das erste Moment der für das Ensemble charakteristischen Verteilungsdichte

$$\overline{v} = \int_{-\infty}^{\infty} v n(v)dv = M_1, \tag{1.10}$$

und die Varianz lässt sich ausdrücken durch die ersten und zweiten Momente

$$\overline{\Delta^2 v} = \int_{\infty}^{\infty} n(v)(v-\overline{v})^2 dv = \int_{\infty}^{\infty} n(v)(v^2 - 2v\overline{v} + \overline{v}^2)dv = M_2 - M_1^2 = \overline{v^2} - \overline{v}^2. \tag{1.11}$$

Existieren alle positiv ganzzahligen Momente, dann ist dadurch die Verteilung eindeutig festgelegt (Rekonstruktionstheorem).

Eine Dichte, die viele empirische Messdaten gut beschreibt, ist die Gaußverteilung

$$n(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{1}{2}\frac{(x-\overline{x})^2}{\sigma^2}}, \tag{1.12}$$

die in der folgenden Abbildung dargestellt ist.

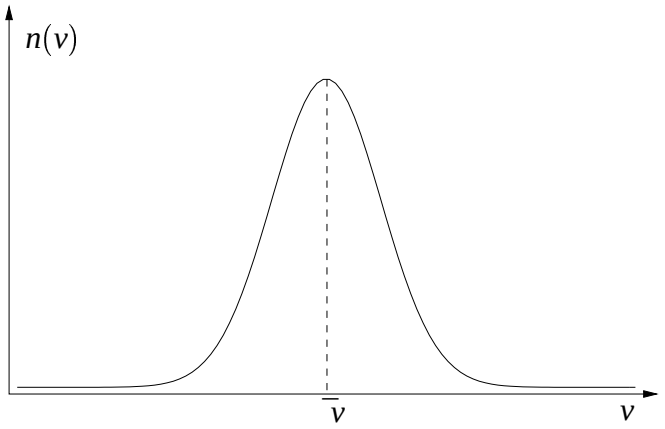


Abb. 1.2 *Gaußverteilung*

Bei der Gaußverteilung liegen bereits durch Mittelwert und Varianz die gesamte Verteilung und damit auch alle höheren Momente fest.

Per Definition ist jede Verteilungsdichte normierbar, d.h. $M_0 = 1$ existiert immer. Allerdings müssen nicht immer alle höheren Momente existieren. Insbesondere kann es vorkommen, dass die Integralausdrücke für Varianz oder Mittelwert divergent sind und M_1 bzw. M_2 demzufolge nicht existieren. Dies ist typisch für Potenzgesetzverteilungen der Form

$$n(x) \propto x^{-\alpha}. \tag{1.13}$$

Damit die Verteilung normierbar ist, muss immer $x > 1$ sein. Das $p-$te Moment ist

$$M_p \propto \int_1^\infty x^{-\alpha+p} \propto \frac{1}{-\alpha+p+1}, \quad \text{falls } -\alpha+p+1<0, \tag{1.14}$$

und ist damit nur dann endlich, wenn $\alpha > p+1$. Die Varianz exsistiert daher nur für Verteilungen mit $\alpha > 3$. Beispielsweise hat die Lorentz-Verteilung $n(x) \propto 1/(1+x^2)$ mit einem asymptotischen $\alpha = 2$ keine endliche Varianz.

Ein Beispiel für eine Verteilung, bei der bereits der Mittelwert nicht mehr existiert, ist das Gutenberg-Richter-Gesetz $n(E) \sim E^-\alpha$ für die Häufigkeit von Erdbeben einer Energie E, bei dem $\alpha \approx 1$. Die Normierbarkeit $(p = 0)$ bleibt in einem solchen Fall nur dadurch gewährleistet, dass ab einem (bisher noch nicht beobachteten) "cutoff"-Wert E die Wahrscheinlichkeit eines Bebens stärker als mit $\alpha = 1$ gegen 0 strebt.

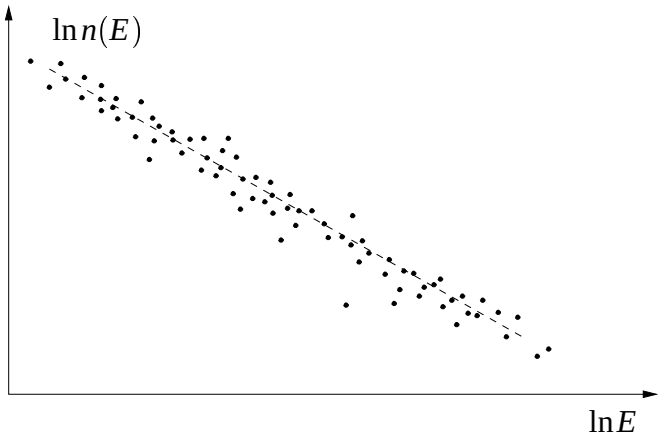


Abb. 1.3 *Gutenberg-Richter-Verteilung der Häufigkeit von Erdbeben einer bestimmten Energie*

Natürlich sind Mittelwert und Varianz endlicher Messreihen immer endlich. Man erkennt die Divergenz eines Momentes in der Regel daran, dass dieses systematisch anwächst, wenn die Messreihe verlängert wird.

1.1.3 ξ^2-Verteilung

Sollte die Verteilungsdichte eines Ensembles etwa aus theoretischen Überlegungen einer Gauß- oder im Prinzip auch einer anderen Verteilung entsprechen und nehmen wir weiter an, dass diese einen theoretischen Mittelwert $\overline{x}$ bzw. eine Varianz σ^2 gekennzeichnet ist, dann kann durch eine Untersuchung der Verteilung der quadratischen Abweichungen entschieden werden, ob diese theoretischen Überlegungen tatsächlich zutreffen. Dazu bestimmt man für eine gegebene Messreihe $x_i, i = 1 \ldots N$ den Wert

$$\xi^2 = \frac{1}{\sigma^2} \sum_i^N (x_i - \overline{x})^2 \tag{1.15}$$

Die Verteilung der ξ^2 für mehrere gleichlange Messreihen hängt sowohl von der Anzahl der Messwerte N als auch von der zugrundeliegenden wirklichen Verteilung ab. In der Regel wird es sich dabei um eine Verteilung handeln, die um N herum ein scharfes Maximum aufweist, denn man wird erwarten, dass die Summe über N Abweichungsquadrate etwa dem N-fachen der Varianz entspricht.

Ist die zugrundeliegende Verteilung die Gaußverteilung und sind gleichzeitig die Messwerte statistisch unabhängig, dann folgt ξ^2 der Verteilung

$$n(\xi^2) = \frac{(\xi^2)^{\frac{N}{2}-1} e^{-\frac{\xi^2}{2}}}{2^{\frac{N}{2}} \Gamma(\frac{N}{2})}, \tag{1.16}$$

wobei die Γ-Funktion durch

$$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt, \quad \Gamma(n+1) = n!, \; n \in \mathbb{N} \tag{1.17}$$

gegeben ist. Liegt ξ^2 zu weit von dem Peak dieser Verteilung entfernt, dann ist mindestens eine der Voraussetzungen nicht erfüllt.

Hilfreich ist eine Untersuchung der ξ^2 häufig bei Prozessen, die gleichverteilte Ergebnisse liefern sollten (etwa die Lottoziehung oder ein Zufallszahlengenerator). Jedes Intervall wird mit einer bestimmten Wahrscheinlichkeit p getroffen. Falls die Zahl der Ziehungen N_z genügend groß ist, ist die Anzahl der Treffer t_i in einem Intervall gaußverteilt mit $\overline{t} = N_z p$ und $\sigma^2 = N_z p(1-p)$ [1]. Ist p für alle Intervalle gleich, dann können wir ein ξ^2 durch Summation der quadratischen Abweichungen der Treffer in jedem Intervall ermitteln [Eq. (1.16)]. Einsetzen in (1.16) ergibt eine "Wahrscheinlichkeit". Wir können dann die Lottoziehung als gerecht oder den Zufallszahlengenerator als gut bezeichnen, wenn dieser Wert groß genug ist.

Weiterhin lässt sich prüfen, ob die Fehler, die bei dem Fit einer Funktion an empirisch ermittelte Daten auftreten, rein statistischer Natur sind, oder ob etwa systematische Abweichungen vorhanden sind (vgl. Press et al., *Numerical Recipes*, Cambridge.)

1.1.4 Fehlerfortpflanzung, Fehler des Mittelwertes

Eine wichtige Frage ist die nach den Fehlern, mit denen empirisch bestimmte Momente wie beispielsweise der Mittelwert verbunden sind. Hier und in anderen Fällen hilft das Fehlerfortpflanzungsgesetz. Seien x, y Größen, deren Schwankungsbreite σ_x, σ_y bekannt ist (etwa bestimmt aus der Varianz einer Messreihe). Sei weiter $f(x,y)$ eine differenzierbare Funktion dieser Größen. Wenn σ_x, σ_y klein genug und die x, y statistisch unabhängig sind, dann ist der Fehler von f

$$\sigma_f = \sqrt{\left(\frac{\partial f}{\partial x}\sigma_x\right)^2 + \left(\frac{\partial f}{\partial y}\sigma_y\right)^2}. \tag{1.18}$$

Diese Beziehung verallgemeinert sich für eine beliebige Anzahl unabhängiger Variablen. Etwa ist für den Mittelwert

$$f(x_1, x_2, \ldots, x_N) = \frac{1}{N}\sum_i x_i. \tag{1.19}$$

Wenn wir berücksichtigen, dass jeder Wert den gleichen Fehler σ_x aufweist und die partielle Ableitung nach den x_i für alle i gleich $1/N$ ist, dann ergibt sich der Fehler des Mittelwertes zu

$$\sigma_{\overline{x}} = \sqrt{\sum_i \left(\frac{1}{N}\sigma_x\right)^2} = \sqrt{\frac{1}{N}\sigma_x^2} = \sigma_x/\sqrt{N} \tag{1.20}$$

Diese Beziehung kann man als Faustregel für die Genauigkeit von Simulationen verwenden, bei denen Ensemblemittelwerte gebildet werden müssen. Um den Fehler an einer Größe um den Faktor 10 zu reduzieren, müssen in der Regel um den Faktor 100 mal mehr Konfigurationen ausgewertet (Messwerte aufgenommen) werden.

1.2 Zeitreihen

Gehen wir auf das einführende Beispiel des Straßenverkehrs zurück und betrachten wir jetzt die Geschwindigkeit eines einzelnen Autos, das die Straße entlang fährt und dessen Tacho kontinuierlich abgelesen wird. Die Geschwindigkeit ist eine kontinuierliche Funktion der Zeit. Aus praktischen (Begrenzung der Datenmenge) oder messtechnischen Gründen (Reaktionszeit eines Instrumentes) wird diese Funktion in aller Regel durch eine endliche Anzahl von Messwerten repräsentiert, die nach bestimmten Regeln zu bestimmten Zeitpunkten aufgenommen werden. Die Folge dieser Messwerte $v_i,\ i = 1 \ldots N$ nennen wir eine *Zeitreihe*.

1.2.1 Zeitmittel

Wir definieren den Zeitmittelwert einer kontinuierlichen Größe $v(t)$ als

$$\overline{v(t)} = \lim_{T\to\infty} \frac{1}{T} \int_0^T v(t)dt. \tag{1.21}$$

Werden bei einer Zeitreihe Messwerte in festen zeitlichen Abständen Δt gebildet, zwischen denen sich der Messwert nur wenig ändert, so kann der Mittelwert der Zeitreihe als Approximation des Zeitmittels verwendet werden:

$$\overline{v(t)} \approx \frac{1}{T} \int_0^T v(t)dt \approx \frac{1}{T} \sum_{i=1}^{T/\Delta t} v_i \Delta t = \frac{\Delta t}{T} \sum_{i}^{T/\Delta t} v_i = \frac{1}{N} \sum_{i}^{N} v_i. \tag{1.22}$$

1.2.2 Abhängigkeit des Mittelwertes von der Wahl der Messzeitpunkte

Dieser Zusammenhang ist nicht für jede Wahl der Messzeitpunkte verallgemeinerbar. Stellen wir uns etwa vor, dass wir die Geschwindigkeit nicht in festen Zeitintervallen ablesen, sondern uns an den Leitpfosten am Strassenrand orientieren und immer nach Durchfahren eines festen Streckenintervalles Δs ablesen. Die gemessene Folge von Messwerten sei v_i^s. Der Zeitmittelwert der Geschwindigkeit ist gleich der Gesamtstrecke dividiert durch die benötigte Zeit,

$$\overline{v}^t = \frac{\sum_i \Delta s}{\sum_i \frac{\Delta s}{v_i^s}} = N \frac{1}{\sum_i \frac{1}{v_i^s}} = \left(\frac{1}{N} \sum \frac{1}{v_i^s} \right)^{-1} \tag{1.23}$$

und damit sicherlich ein anderer Wert als das arithmetische Mittel der Messungen $\overline{v}^s = (1/N) \sum_i v_i^s$.

1.3 Konfigurations- und räumliche Mittelwerte

Als Konfiguration eines Systems bezeichnet man die Menge der Orts- und, falls relevant, der Impulskoordinaten der Teilchen eines Systems. Ein Konfigurationsmittelwert

ist ein Mittelwert über die Teilchen und mehrere (unabhängige) Konfigurationen eines Systems.

1.3.1 Ergodenhypothese

Es ist eine wichtige Frage, wann Zeit- und Konfigurationsmittelwerte trotz der verschiedenen Bestimmungsvorschriften übereinstimmen. Auf dem Computer ist es nämlich wie im Experiment häufig einfacher, ein System in der Zeit zu verfolgen als statistisch unabhängige Konfigurationen zu generieren. Die statische Physik ermittelt Größen durch Integration über den Phasenraum und Wichtung mit der für das Ensemble charakteristische Verteilungsfunktion, d.h. sie bildet den Konfigurationsmittelwert. In den weitaus meisten praktischen Fällen stimmen die beiden Mittelwerte tatsächlich überein, allgemein zeigen lässt sich das jedoch nicht. Die angenommene Übereinstimmung ist Gegenstand der *Ergodenhypothese*.

Für unsere Straße kann man sich vorstellen, dass man alle möglichen Verkehrssituationen (Engstellen, Geschwindigkeitsbeschränkungen) erfasst, gleich ob man mit einem Auto die Strecke abfährt oder viele hintereinander fahrende Autos benutzt, deren Geschwindigkeitsinformation aber nur als Schnappschuss ausgewertet wird. Damit wäre dann das Konfigurations- gleich dem Zeitmittel.

Umgekehrt wird die Ergodenhypothese sicher verletzt sein, wenn das verfolgte Auto nicht die gesamte zur Verfügung stehende Strecke abfährt (Straßensperre). In physikalischen Systemen kann das etwa durch hohe Energiebarrieren zwischen Zuständen bewirkt werden. Als Beispiele seien hier Spingläser oder der Zellularautomat “Q2R” genannt (vgl. Abschnitt 5.4.1).

1.3.2 Räumlich und zeitlich lokale Mittel

Räumliche Mittel werden eingeführt, um von Situationen mit diskreten Variablen zu einer Beschreibung des Systems durch kontinuierliche Variablen zu gelangen. Im Verkehrsbeispiel spricht man gerne von Verkehrsfluß (Anzahl der Autos pro Zeiteinheit) oder Verkehrsdichte (Anzahl der Autos pro Streckenabschnitt). Wir zählen dazu in der Regel die Teilchen/Autos in einem bestimmten Streckenabschnitt Δs um eine gewählten Ort $\vec{x}$ herum. Um den Fluss $\Phi(\vec{x},t)$ zu bestimmen, wählen wir einen festen Ort aus und bestimmen die Anzahl N_t der Fahrzeuge, die den Ort innerhalb dieses Intervalls Δt passieren.

$$\Phi_t = \frac{N_t}{\Delta t} \tag{1.24}$$

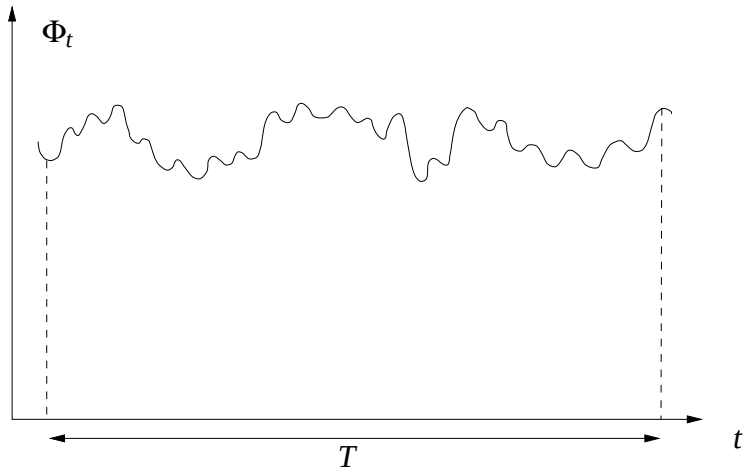


Abb. 1.4 *Typischer Verlauf des Flusses als Funktion der Zeit*

In beiden Fällen, sowohl der räumlich als auch der zeitlich lokalen Mittelung muss man einen Kompromiss zwischen der statistischen Genauigkeit und der Lokalität eingehen. Je genauer das Mittel bestimmt werden soll, desto größer ist das räumliche oder zeitliche Intervall, das dazu festgelegt werden muss und desto geringer ist die räumliche und zeitliche Auflösung der betrachteten Größe.

Durch entsprechende räumlich oder zeitlich lokale Mittelung lässt sich natürlich auch eine lokale Durchschnittsgeschwindigkeit ermitteln. Nehmen wir an, es gäbe Verkehrsstaus. Dann wird sich das in der lokalen Geschwindigkeit niederschlagen

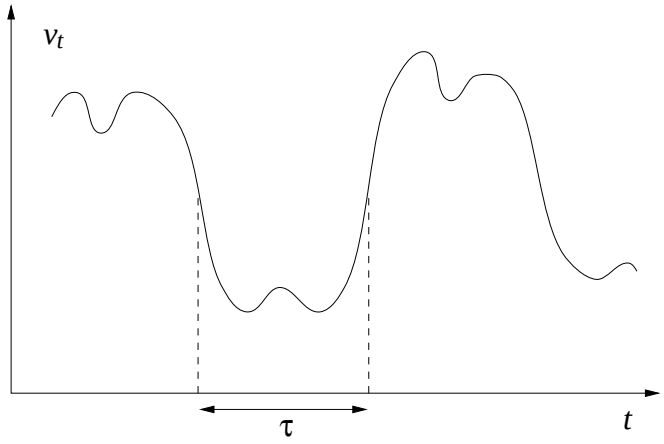


Abb. 1.5 *Möglicher Zeitverlauf der Geschwindigkeiten*

und die momentane Dichte hätte dann typischerweise eine solche Form:

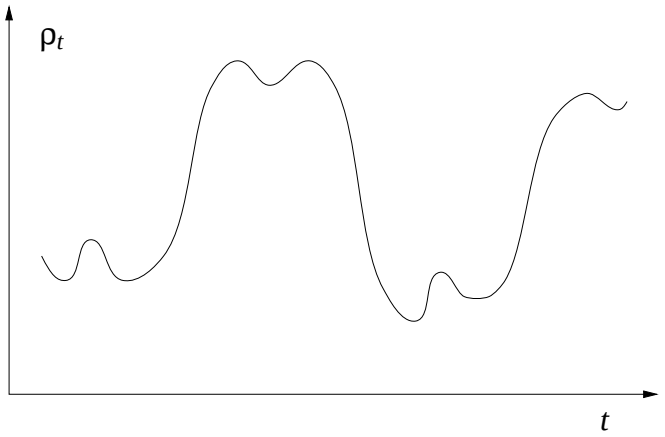


Abb. 1.6 *Momentane Dichte*

Diese Strukturen kann man möglicherweise durch eine charakteristische Dauer τ und eine charakteristische Länge ξ beschreiben, die etwa der Dauer und Länge des Staus entsprechen können. Nennen wir die gesamte Beobachtungsdauer T und die Länge des gesamten beobachteten Straßenabschnitts L. Es ist klar, dass dann

$$T \gg \tau \gg \Delta t \quad \text{sowie} \quad L \gg \xi \gg \Delta s \tag{1.25}$$

gelten muss, damit die gebildeten Mittelwerte sinnvolle physikalische Aussagen enthalten. Man beachte, dass diese Bedingung, die man auch als *Skalenseparation* bezeichnet, möglicherweise nicht immer erfüllt werden kann.

1.4 Statistische Unabhängigkeit, Korrelationen

Zwei Größen A und B heißen genau dann statistisch unabhängig, wenn $\overline{AB} = \overline{A}\,\overline{B}$ gilt.

Um diese Aussage anschaulich zu machen, betrachten wir die Geschwindigkeiten $v_i^{0)}$ und $v_i^{(1)}$ aller Autos i, die sowohl den Ort x_0 als auch x_1 passieren. Wir schreiben die Geschwindigkeiten als Mittelwerte plus Abweichung,

$$v_i^{(0)} = \overline{v^{(0)}} + \Delta v_i^{(0)}, \quad v_i^{(1)} = \overline{v^{(1)}} + \Delta v_i^{(1)}. \tag{1.26}$$

Jetzt bilden wir das Produkt

$$\begin{aligned}
\overline{v_i^{(0)} v_i^{(1)}} &= \overline{\left(\overline{v^{(0)}} + \Delta v_i^{(0)}\right)\left(\overline{v^{(1)}} + \Delta v_i^{(1)}\right)} && (1.27)\\
&= \overline{\overline{v_i^{(0)}}\;\overline{v_i^{(1)}}} + \overline{\overline{v^{(0)}}\Delta v_i^{(1)}} + \overline{\Delta v_i^{(0)}\overline{v^{(1)}}} + \overline{\Delta v_i^{(0)}\Delta v_i^{(0)}} && (1.28)\\
&= \overline{\overline{v_i^{(0)}}\;\overline{v_i^{(1)}}} + \overline{\Delta v_i^{(0)}\Delta v_i^{(0)}} && (1.29)
\end{aligned}$$

Es ist klar, dass die Mittelwerte der Produkte aus einem Mittelwert und den Fluktuationen verschwinden müssen, denn der Mittelwert selbst ist eine Konstante und der

Mittelwert der Fluktuationen per Definition gleich Null. Es verbleibt also nur das Produkt der Mittelwerte und der Mittelwert des Produktes der Fluktuationen. Vereinfacht gesprochen verschwindet letzterer nur dann (proportional zu $1/\sqrt{N}$), wenn das Produkt genauso häufig positive wie negative Werte annimmt. Das kann es nur, wenn die Vorzeichen der Fluktuationen genauso häufig gleich wie verschieden sind.

Im Strassenbeispiel könnten x_0 etwa einen Punkt am Gipfel eines Hügels und x_1 einen auf dem Gefälle kennzeichen. Ein "Raser" produziert wahrscheinlich sowohl bei x_1 als auch x_2 positive Abweichungen und ein "Sonntagsfahrer" beide Male negative Δv_i. In diesem Fall sind die Geschwindigkeiten korreliert. Sogar wenn wir unterwegs die zwei Fahrer irgendwie austauschen, wird durch das Zusammtreffen jeweils einer positiven mit einer negativen Abweichung immer eine Korrelation vorliegen. Erst wenn wir die Fahrer mit Wahrscheinlichkeit 50% austauschen und ansonsten weiterfahren lassen, hängen die beiden Abweichungswerte nicht mehr vorhersagbar miteinander zusammen. Nur dann erhalten wir unkorrelierte Geschwindigkeiten.

Als Maß für die Korrelation führen wir eine zwischen -1 und 1 liegende Zahl ein:

$$-1 \leq \frac{\overline{AB} - \overline{A}\,\overline{B}}{\Delta A \Delta B} \leq 1, \quad \Delta A = \sqrt{\overline{\Delta^2 A}}, \Delta B = \ldots \tag{1.30}$$

Die Gültigkeit dieser Beziehung folgt aus der Schwarz'schen Ungleichung, die besagt, dass das Skalarprodukt zweier Vektoren betragsmäßig immer kleiner ist als das Produkt ihrer Normen, für den Zähler der obigen Beziehung also besagt

$$\begin{aligned} |\overline{AB} - \overline{A}\,\overline{B}| &= |\overline{\Delta A_i \Delta B_i}| && (1.31)\\ &= (1/N)\,|\sum_i^N \Delta A_i \Delta B_i| && (1.32)\\ &\leq (1/N)\sqrt{\sum_i \Delta A_i}\sqrt{\sum_i \Delta B_i} && (1.33)\\ &= \Delta A\,\Delta B. && (1.34) \end{aligned}$$

Die ersten zwei Zeilen sind dabei nichts anderes als unsere vorhergegangene Überlegung, die dritte ist die Aussage der Schwarz'schen Ungleichung und die vierte benutzt die Definition von ΔA und ΔB. Das hier verwendete Skalarprodukt ist das euklidische im $\mathbb{R}^N$.

Wir verallgemeinern die Beziehung (1.30) indem wir beispielsweise die Geschwindigkeiten eines einzelnen Autos zu einer beliebigen Zeit t und einem späteren Zeitpunkt $t+\tau$ miteinander korrelieren,

$$C_v(\tau) = \frac{\overline{v(t)v(t+\tau)} - \overline{v(t)v(t+\tau)}}{\Delta v(t)\Delta v(t+\tau)} \tag{1.35}$$

Dem Mittelwert haben wir einen Sinn durch Zeitmittelung über alle Anfangszeitpunkte t gegeben. Das macht natürlich nur dann Sinn, wenn das betrachtete System zeittranslationsinvariant ist (d.h. unser Fahrer darf nicht ermüden und die Straße muss eben sein

oder zumindest überall gleich hügelig). Dann sind die Mittelwerte und Fluktuationen von $v(t)$ und $v(t+\tau)$ gleich. Wir erhalten durch Einsetzen der Integraldefinition für den Zeitmittelwert

$$C_v(\tau) = \frac{\overline{v(t)v(t+\tau)} - \overline{v(t)}^2}{\Delta^2 v(t)} = \frac{\frac{1}{T}\int_0^T (v(t+\tau)-\overline{v})(v(t)-\overline{v})dt}{\frac{1}{T}\int_0^T (v(t)-\overline{v})^2 dt} \tag{1.36}$$

Diese *Autokorrelationsfunktion* $C_v(\tau)$ sagt z.B. etwas über das Fahrverhalten aus (Auto=Selbst≠Fahrzeug). Ein nervöser Fahrer zuckt häufig mit dem Fuß vom Gaspedal und $c_v(\tau)$ geht schnell gegen 0. Der notorische Linksfahrer mit Bleifuß hingegen produziert ein langsam abnehmendes $C_v(\tau)$. Als Korrelationszeit der Geschwindigkeit τ_v definiert man häufig den Wert $\int_0^\infty C_v(\tau)d\tau$ wenn der existiert. Sie ist sozusagen ein Maß für die typische Zeit zwischen zwei signifikanten Bewegungen des Fußes auf dem Gaspedal. In speziellerem Kontext verwendet man auch die Nullstelle der Tangenten bei $(\tau = 0, C_v(0) = 1)$, als Korrelationszeit.

Häufig fällt die Autokorrelation einer Größe A exponentiell mit der Zeit ab

$$C_A(t) \approx \exp(-\frac{t}{\tau_A}) \tag{1.37}$$

so dass die Defintionen über die Tangente bei $\tau = 0$ und die Integraldefinition der Korrelationszeit den gleichen Wert ergeben. Manchmal ist der Abfall von $C_A(t)$ nicht exponentiell, sondern folgt z.B. einem Potenzgesetz

$$C_A(t) \propto t^{-z} \tag{1.38}$$

Für $z \geq 1$ konvergiert dieses Integral natürlich nicht. Dieser Fall signalisiert häufig ein interessantes physikalischen Verhalten, das von dem üblichen abweicht und kann manchmal mit so genannten *kritischen Phänomenen* in Verbindung gebracht werden (darüber später mehr, etwa im Kapitel über Perkolation).

Die Relaxation eines Systems in ein Gleichgewicht läßt sich durch die Nichtgleichgewichts-Autokorrelationsfunktion beschreiben,

$$C_A^0(\tau) = \frac{\langle A(0)A(\tau)\rangle - \langle A^2\rangle}{\langle \Delta^2 A\rangle} \tag{1.39}$$

und die Nicht-Gleichgewichtsrelaxationszeit dann als

$$\tau_A' = -\int_0^\infty C_A(t)dt \tag{1.40}$$

definieren. Da wir hier einen Startzeitpunkt festlegen, sind wir gezwungen, ein Ensemblemittel $\langle \ldots \rangle$ durchzuführen.

Schließlich kann man noch eine räumliche Korrelationsfunktion definieren durch

$$C_A(x) = \frac{[A_j A_{j+x}] - [A]^2}{[A^2] - [A]^2} \tag{1.41}$$

mit der charakteristischen Länge

$$\xi_A = \int_0^\infty C_A(x)dx \tag{1.42}$$

wobei wir hier die eckigen Klammern für ein lokales räumliches Mittel verwendet haben.

Kapitel 2

Zufallszahlen

Um in einem Modell die Unvollständigkeit unseres Wissens über die Vorgänge auf Skalen auszudrücken, die kleiner als die Modellierungsskala sind (Brown'sche Bewegung, Rauschen in elektronischen Systemen ...), benutzt man üblicherweise eine stochastische Komponente. Viele der Modelle, Wir werden in den folgenden Kapiteln viele solcher Modelle kennenlernen. Zu deren Implementierung auf dem Computer brauchen wir daher eine "Zufalls"quelle, in der Regel ein so genannter Pseudozufallszahlengenerator, der "zufällige" Zahlen in einem bestimmten Bereich liefert.

Nun ist ein Computer von seiner Natur her eine deterministische Maschine und alle Versuche, Zufallszahlen per Software zu generieren, sind dazu verdammt, uns zufällig "aussehende" Sequenzen ansonsten deterministisch erzeugter Zahlen zu liefern. In der Natur hingegen finden wir "wirklich"zufällige Sequenzen, etwa dann, wenn wie beim radioaktiven Zerfall quantenmechanische Effekte ins Spiel kommen, die wir prinzipiell nicht beeinflussen können. Wir ziehen uns also auf eine Turing-artige Definition von (Pseudo)Zufallszahlen zurück und fordern, dass sich "von außen" die stochastischen Eigenschaften beider Klassen nicht unterscheiden sollen. Wie wir sehen werden, wird sich dieser Anspruch nur in einem begrenzten Umfang erfüllen lassen.

2.1 Einführung

Neben die statistischen Anforderungen an eine Zufallssequenz treten auf einem Computer weitere, die für die Einsetzbarkeit in Simulationsprogrammen wichtig sind. Hierzu zählen (i) die Reproduzierbarkeit der Sequenz, die zum Testen von Programmen wichtig ist. Diese umfasst einerseits die Abhängigkeit der Sequenz von einem vorgebbaren Ausgangszustand und andererseits die Portierbarkeit des Erzeugungsmechanismusses auf verschiedene Rechnerarchitekturen (wenn man etwa ein auf einer Workstation entwickeltes Programm später auf einem Parallelrecher laufen lassen möchte). Wichtig ist (ii) auch die Effizienz des Generators, denn in vielen Fällen, z.B. der Auswertung hochdimensionaler Integrale mit Monte-Carlo Methoden, ist die Ermittlung von Zufallszahlen ein bedeutender Anteil des Gesamtaufwandes des Programms.

Wir werden uns zunächst auf Generatoren beschränken, die ganze Zahlen zwischen 0 und einem Maximalert p liefern. Die Zahlen sollen idealerweise in diesem Intervall gleichverteilt sein. Am Ende dieses Kapitels soll dann besprochen werden, wie wir auch Zufallszahlen mit einer anderen als der Gleichverteilung erzeugen können. Im Sinne unserer Definition wollen wir von diesen Zahlen fordern, dass sie keine statistischen Korrelationen aufweisen, d.h. dass für eine Sequenz x_i solcher Zufallszahlen die Beziehung

$$\langle x_i x_{i+n} \rangle = \langle x_i \rangle^2 \tag{2.1}$$

gelten soll, unabhängig von der Verschiebung n. Der Mittelwert versteht sich dabei über die Länge der Sequenz bzw. auch über mehrere Sequenzen.

Zusammenfassend fordert man für einen guten Zufallszahlengenerator,

1. dass die gelieferten Zahlen einer vorgegebenen Verteilung, meist einer Gleichverteilung folgen,
2. dass Korrelationen verschwinden sollen,
3. damit insbesondere, dass sich die Zufallszahlen nicht früher wiederholen, als durch eine bekannte, genügend große Zahl vorgegeben; in der Regel soll die *Periode* des Generators mindestens 10^{11} Zahlen umfassen,
4. Schnelligkeit,
5. Reproduzierbarkeit der Zufallszahlenfolge zu Testzwecken.

Zufallszahlen zwischen 0 und p lassen sich durch Division mit p auf das (ggf. offene) Intervall $[0,1]$ abbilden. Der Anwendungsfall entscheidet darüber, ob das offene oder abgeschlossene Intervall praktischer ist.

Übung: man überlege sich, wie sich aus Zufallszahlen aus dem Intervall $[1, p-1]$ Zufallszahlen aus einem anderen Intervall, etwa $[0,q]$ erzeugen lassen, die wiederum möglichst gut gleichverteilt sind (etwa stellvertretend für Lottozahlen). Um mögliche Probleme zu sehen, überlegen Sie sich, ob Ihr Verfahren Zahlen bevorzugt oder gar nicht erst erzeugt, wenn $q > p$ ist oder geringfügig kleiner als p.

Im folgenden sollen Generatoren für homogen verteilte ZZ betrachtet werden. Die beiden wichtigsten Typen von Zufallszahlengeneratoren sind (i) *kongruentielle* und (ii) *lagged Fibonacci*-Generatoren.

2.2 Kongruentielle Generatoren

Kongruentielle Generatoren wurden zuerst von Lehmer (1948) vorgeschlagen. Man erhält eine Sequenz positiver ganzer Zufallszahlen x_i durch die Rekursionsvorschrift

$$x_{i+1} = (cx_i) \bmod p, \quad c, p \in \mathbb{Z} \tag{2.2}$$

Hier bedeutet die Modulo-Operation, dass der positive Rest der Division mit p gebildet werden soll. Mit anderen Worten ziehen wir von der linksstehenden Zahl das Ergebnis der ganzzahligen Division multipliziert mit der rechtstehenden Zahl ab:

$$n \bmod m = n - \left\lfloor \frac{n}{m} \right\rfloor m, \tag{2.3}$$

wobei die eckigen (Gauß-) Klammern die größte ganze Zahl bedeuten sollen, die gerade kleiner als (n/m) ist.

Der Anfangswert x_0, Keim oder *Seed* genannt, charakterisiert die Sequenz vollständig. Der spezielle Wert $x_0 = 0$ perpetuiert sich, und wir lassen ihn daher nicht als Anfangswert zu. Die maximal mögliche Periode eines solchen Generators ist $p-1$, genau dann wenn jeder der $p-1$ Reste von 1 bis $p-1$ in der Sequenz auftaucht. Sobald sich ein Rest wiederholt, wiederholt sich ab dort gemäß (2.2) die gesamte Sequenz.

Wichtig ist die Frage, wann die Periode gleich $p-1$ wird, also alle möglichen Restwerte wirklich angenommen werden. Die folgende Aussage dazu geht auf Carmichael (1910) zurück. Die Periode ist dann maximal, wenn (i) p eine Primzahl ist und gleichzeitig (ii) c so gewählt ist, dass p die kleinste Zahl ist, für die gilt $c^{p-1} \bmod p = 1$. Wir beweisen diese Behauptung in zwei Schritten.

Sei zunächst angenommen, dass p keine Primzahl ist. Dann lässt sich p schreiben als Produkt zweier Zahlen $p_1 p_2$, die beide nicht notwendigerweise prim sein müssen. Nehmen wir jetzt an, die Periode sei maximal. Dann können wir beispielsweise o.B.d.A. x_i als Vielfaches np_1 von p_1 wählen. Für x_{i+1} erhalten wir

$$x_{i+1} = cnp_1 \bmod p = cnp_1 - \left\lfloor \frac{cnp_1}{p_1 p_2} \right\rfloor p_1 p_2 = (cn - \left\lfloor \frac{cn}{p_2} \right\rfloor p_2) p_1. \tag{2.4}$$

Da der Ausdruck in runden Klammern eine ganze Zahl ist, ist das Ergebnis wieder ein Vielfaches von p_1. Damit sind weiterhin auch alle weiteren x_i Vielfache von p_1 und durchlaufen damit nur eine Teilmenge aller möglichen Restwerte. Die Periode kann daher nicht maximal sein.

Um den zweiten Teil der Behauptung zu zeigen, betrachten wir eine Sequenz, die mit der 1 beginnt,

$$\begin{aligned}
x_1 &= 1 = c^0 \bmod p \\
x_2 &= c^1 \bmod p = c - [c/p]p \\
x_3 &= c(c - [c/p]p) \bmod p = c^2 \bmod p \\
&\vdots \qquad (2.5) \\
x_n &= c(c^{n-2} - [c^{n-2}/p]p) \bmod p = c^{n-1} \bmod p \\
&\vdots \qquad (2.6) \\
x_p &= c^{p-1} \bmod p = 1
\end{aligned}$$

Oben haben wir benutzt, dass $[c^k/p]p$ ein Vielfaches von p ist, so dass $[c^k/p]p \bmod p = 0$. Sobald der Rest 1 wieder auftritt, wiederholt sich die Folge. Damit

die Periode maximal wird, darf dass erst bei dem p-ten Glied der Fall sein. Damit muss dann $p-1$ die kleinste Zahl sein, für die $c^{p-1} \bmod p = 1$ gilt (qed). Die Einschränkung auf $x_1 = 1$ kann aufgegeben werden: man sieht dann, dass sich das Anfangsglied als Vielfaches durch alle Gleichungen hindurchschleift und letztlich die gleiche Aussage resultiert.

2.2.1 Implementierung von Zufallszahlengeneratoren

Für Digitalrechner sind so genannte Mersenne'sche Primzahlen besonders interessant, die sich in der Form $2^n - 1$ darstellen lassen. Der größte ganzzahlige Wert, der bei den meisten 32-Bit Computern in einer `int` Variablen speicherbar ist, gerade die Binärzahl, bei der das erste Bit $= 0$ und alle anderen 31 gesetzt sind, also gerade $2^{31} - 1$. Glücklicherweise liegt für $n = 31$ gerade eine Mersenne'sche Primzahl vor.

Park und Miller (1988) schlagen einen kongruentiellen Generator mit $p = 2^{31} - 1$ und $c = 7^5 = 16807$ vor. Man könnte versuchen, diesen in der folgenden Weise zu implementieren:

```
const int p = 2147483647;
const int c = 16807;

int   rnd   = 42; // seed
//...
rnd = (c * rnd) % p; // Problem
```

Allerdings verlässt das Produkt `c * rnd` für große `rnd` den zulässigen Zahlbereich. Um jetzt nicht den Wert für p und damit die Periodenlänge einschränken zu müssen, greifen auf einen Trick von Schrage (1979) zurück. Wir suchen positive Zahlen q und r mit $r < q$, so dass $p = cq + r$ gilt, also $q = [p/c]$ und $r = p \bmod q$. Damit gilt (ohne Beweis)

$$cx_i \bmod p = \begin{cases} \underbrace{c(x_i \bmod q) - r[x_i/q]}_{(A)}, & \text{falls } A > 0, \\ c(x_i \bmod q) - r[x_i/q] + p, & \text{sonst.} \end{cases} \tag{2.7}$$

Alle die hier auftretenden Teilausdrücke sind kleiner oder gleich p und können in der angegebenen Weise addiert oder subtrahiert werden, ohne Zahlen zu ergeben, die betragsmäßig größer als p wären. Damit ist eine mögliche portable Implementierung

```
const int p = 2147483647;
const int c = 16807;
const int q = 127773;
const int r = 2836;

int   rnd   = 42; // seed
```

```
//...
// integer Division x_i/q:
int   quot  = rnd/q;
// hier ausgenutzt, um mod einzusparen
rnd         = c * (rnd - q * quot) - r * quot;
// fuer negative Resultate muss p addiert werden
if ( rnd < 0 ) rnd += p;
```

Weitere getestete Wertepaare (p,c) finden sich in Press et al. *Numerical Recipes* [14], Kapitel 7.

"Quick and dirty" Generatoren

In vielen älteren Arbeiten sind Zufallszahlengeneratoren eingesetzt worden, die benutzen, dass man beim Überlauf einer `int` oder `unsigned` Variablen die Modulo-Operation bezüglich der maximal darstellbaren Zahl plus 1 quasi geschenkt bekommt und damit keine zeitaufwändige Division durchgeführt werden muss. Ein typisches Beispiel ist der so genannte IBM-Generator, hier in FORTRAN77 implementiert

```
c     4 byte integer
      integer*4    ibm
c      ...
      ibm = ibm * 65539
      if(ibm.lt.0) ibm = ibm + 2147483647 + 1
```

Man beachte, dass beim Überlauf die Modulo-Operation bezüglich 2^{31} durchgeführt wird und nicht bezüglich $2^{31}-1$; dieser Generator hat also mit Sicherheit eine Periode kürzer als 2^{30}, da 2 als Faktor in p auftaucht. Die Addition von 2^31 muss in zwei Schritten ausgeführt werden, da 2^31 nicht als 4-Byte `integer*4` darstellbar ist. Der IBM-Generator leidet unter starken sequentiellen Korrelationen, wie wir sie im nächsten Abschnitt besprechen werden. Für einen ungeraden Anfangswert werden alle ungeraden Zahlen durchlaufen, das letzte Bit der erzeugten Zufallszahl ist also immer gesetzt. Im Allgemeinen sollte man bei kongruentiellen Generatoren die niederwertigen Bits niemals als unabhängig Zufallsvariablen verwenden, da sie häufig stark korreliert sind.

In C kann man den folgenden, noch geringfügig schnelleren Generator konstruieren, der den Überlauf einer 32-Bit `unsigned` Variablen ausnutzt

```
unsigned rnd; // testen ob 4 byte lang!
...
rnd = 1664525 * rnd + 1013904223;
```

der jedoch die maximal mögliche Periodenlänge von 2^32-1 aufweist (das Vorzeichenbit ist Bestandteil der Zahl). Er gehört zu einer etwas allgemeineren als der hier besprochenen Klasse von Generatoren, die der Rekursionsvorschrift $x_{i+1} = (cx_i + d) \bmod p$

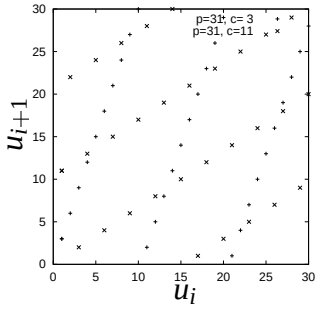


Abbildung 2.1: 2-cube Korrelationen für die Generatoren (31,3) und (31,11)

genügen. Die Zahlen sind abwechselnd gerade und ungerade, das niederwertige Bit also perfekt antikorreliert.

Die beiden eben gezeigten Generatoren sollte man in ernsthaften Simulationen nicht einsetzen. Ebenso sollte man auf die `rand` Funktion der C Bibliothek verzichten, deren ANSI Beispielimplementierung nur eine Periode von weniger als $2^{15}-1$ aufweist. Falls jedoch oberflächliche Tests oder vielleicht ein Videospiel implementiert werden sollen, so sind sie vermutlich eine brauchbare Wahl.

2.2.2 Sequentielle Korrelationen

Eine typische Korrelation, die in kongruentiellen Generatoren auftritt ist, dass auf kleine Zufallszahlen typischerweise wieder kleine Zufallszahlen folgen. Ist etwa in dem Park-Miller "minimal standard" Generator eine Zahl nahe 0 aufgetreten, so wird die nächste vielleicht im Bereich einiger 100 000 liegen, im Vergleich zum Gesamtzahlbereich $2^{31}-1$ also wieder klein sein. Da von Benutzern Anfangswerte häufig kleine Zahlen sind, findet man in der Literatur häufig den Hinweis, man solle den Generator anfangs durch Verwerfen einiger Zufallszahlen "aufwärmen".

Die eben beschriebenen sind ein spezielles Beispiel sequentieller Korrelationen. Allgemein macht man folgenden Test, der seiner Konstruktionsvorschrift gemäß *n*-cube Test genannt wird. Aus der Sequenz $x_i, x_{i+1}, \ldots$ wählt man *n*-Tupel aufeinanderfolgender Zahlen aus, etwa für $n=2$ die Paare (x_i, x_{i+1}), (x_{i+1}, x_{i+2}), (x_{i+2}, x_{i+3}), ... Diese fasst man als Koordinaten in einem *n*-dimensionalen Raum (*n*-cube) auf und trägt sie entsprechend auf. In der folgenden Abbildung ist dies für $n=2$ und zwei verschiedene Generatoren der Periodenlänge 31 durchgeführt: (p,c) ist $(31,3)$ bzw. $(31,11)$.

Man sieht deutlich, dass die Paare auf ausgezeichneten Linien liegen (allgemein auf $n-1$ dimensionalen Hyperebenen), diese aber den Raum $\mathbb{N}\times\mathbb{N}$ nicht dicht ausfüllen, wie man das für wirklich zufällige Zahlen erwartet. Dies kann man einsehen aus dem *Theorem von Marsaglia* (1968), das hier ohne Beweis[1] angegeben sei: Für jedes *n* kann

[1] Als Beweisidee kann man mit $n=1$ beginnen und explizit ein a_0 und a_1 angeben. Multiplikation mit

man mindestens einen Satz Zahlen $a_0 \ldots a_n$, $a_i \in \mathbb{N}^+, a_i < p$ finden, so dass gilt:

$$(a_0 + a_1 c + a_2 c^2 + \cdots + a_n c^n) \bmod p = 0, \tag{2.8}$$

d.h., so dass das angeschriebene Polynom in Klammern ein Vielfaches von p ist. Setzt man anstelle der Potenzen von c nun eine Zufallszahlsequenz ein,

$$\begin{aligned} x_0 &= x_0 \\ x_1 &= cx_0 - [\frac{cx_0}{p}]p \\ &\vdots \\ x_n &= c^n x_0 - [\frac{c^n x_0}{p}]p \end{aligned}$$

so erhält man

$$\begin{aligned} &a_0 x_0 + a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = \\ &\quad (a_0 + a_1 c + a_2 c^2 + \cdots + a_n c^n) x_0 - (a_1 [\frac{cx_0}{p}] + a_2 [\frac{c^2 x_0}{p}] + \cdots + a_n [\frac{c^n x_0}{p}]) p \end{aligned} \tag{2.9}$$

Das geklammerte Polynom im ersten Term auf der rechten Seite war nach Voraussetzung ein Vielfaches von p, und gleiches gilt für den zweiten Term, da der Ausdruck in der runden Klammer eine ganze Zahl ist. Die $a_0, \ldots, a_n$ definieren die Lage einer Hyperebene im Raum und die rechte Seite ist proportional zum Abstand dieser Hyperebene vom Ursprung. Da die Norm der als n-dimensionaler Vektor aufgefassten a_i kleiner ist als $\sqrt{np}$, ist der Abstand der Ebenen größer als $p/\sqrt{np} \sim \sqrt{p/n} > 1$. Damit können die Hyperebenen nicht alle Punkte des $\mathbb{N}^n$ erfassen.

Es lässt sich zeigen, dass sogar maximal $p^{1/n}$ Hyperebenen ausgefüllt werden. Im Falle unseres zweidimensionalen Beispieles mit $p = 31$ also sind 5 Ebenen das mögliche Maximum.

Um sequentielle Korrelationen niedriger Ordnung n zu zerstören, wählt man eine einfache Mischungsvorschrift, die auf Bays und Durham zurückgeht. Man schreibt dazu die ersten gezogenen Zahlen in ein kurzes Feld (typischerweise 32 Elemente). Eine separate Variable speichert die jeweils letzte gezogene Zufallszahl, skaliert sie jedoch in den Bereich der für das Feld zulässigen Indexwerte. Wird jetzt eine neue Zahl gezogen, so benutzt man zunächst diese Variable, um ein Feldelement zu identifizieren, welches als Ergebniszufallszahl verwendet und von der Routine zurückgegeben wird. Danach benutzt man den regulären kongruentiellen Generator, um eine neue Zahl zu würfeln, die man an die jetzt "freigewordene" Position schreibt. Diese Zahl wird jetzt noch in den Indexbereich des Feldes skaliert und dient dann als neuer Positionsanzeiger für die nächste zu ermittelnde Zufallszahl.

---

c, Erhöhung der Indizes um 1 und Einführung einer neuen Konstante a_0 erlaubt einen Induktionsschluss auf $n+1$.

```
// cong_rand() liefere Zufallszahlen im Bereich
//   1 ... MaxRand-1
unsigned  cong_rand();

// Feld und Indexvariable
const  int      N_arr = 32;
static unsigned rnd[N_arr];
static unsigned Idx;

// neuer Generator, hier ohne Initialisierung
unsigned  shuffle(){
   // folgende Zahl geben wir später zurück,
   // (Idx von Initialisierung bzw. vorigem Aufruf bekannt)
   unsigned new_rnd = rnd[Idx];

   // Ersetzen der Zahl auf der benutzten Position
   // neue Zufallszahl ziehen
   unsigned new_cong = cong_rand();
   rnd[Idx] = new_cong;
   // quick and dirty: neues Idx
   // Klammerausdruck ist immer echt kleiner 1
   Idx      = ((double) new_cong / MaxRand) * N_arr;

   return new_rnd;
}
```

2.3 Lagged-Fibonacci Generatoren

Mit dem folgenden Typ von Generatoren, der auf Tausworth (1965) zurückgeht, lassen sich extrem lange Perioden verwirklichen und auch vorteilhafte Aussagen über Korrelationen machen.

Man betrachte eine Sequenz binärer Zahlen $x_i \in \{0,1\}, x_1, x_2, x_3, \ldots x_n$. Im Gegensatz zu unserer bisherigen Notation soll die Zahl x_1 zuletzt generiert worden sein, x_n entsprechend $n-1$ Schritte vorher. Jetzt definieren wir ein neues x_0 durch eine Rekursionsvorschrift

$$x_0 = \sum_i x_{n_i} \bmod 2 \tag{2.10}$$

Die Summe erstreckt sich über eine Untermenge der der Indizes $1, \ldots n$, z.B.

$$x_0 = (x_1 + x_2 + x_5 + x_{18}) \bmod 2 \tag{2.11}$$

Auf Digitalrechnern lässt sich die Addition mod 2 sehr schnell mit Hilfe des ausschliessenden Oders (XOR) implementieren ($\oplus$), das durch durch folgende Wahrheitstafel festgelegt ist

$\oplus$	0	1
0	0	1
1	1	0

Bevor wir diese Ideen zur Implementierung eines Zufallszahlengenerators benutzen, vertiefen wir noch etwas die zugehörige Theorie.

2.3.1 Primitive Polynome

Die Summe (2.10) lässt sich mit einem Polynom assoziieren. Die linke Seite entspricht einer konstanten 1 und jeder in der Summe auftretende Term einer Potenz der Variablen x, die der Position in der Sequenz entspricht. Für das Beispiel (2.11) erhalten wir so

$$f(x) = 1 + x + x^2 + x^5 + x^{18}. \tag{2.12}$$

Für eine Bitsequenz aus 18 Bit, bei der nach Vorschrift (2.10) jeweils das jüngste Bit ersetzt wird, können insgesamt 2^{18} verschiedene Muster generiert werden. Die folgende Aussage sichert jetzt, dass die Rekursionsvorschrift tatsächlich alle dieser Möglichkeiten erfasst: Die Periode der Vorschrift (2.10) ist genau dann maximal, wenn das assoziierte Polynom $f(x)$ primitiv ist, d.h. über dem durch binäre Multiplikation und Addition erzeugten Körper nicht faktorisiert werden kann.

Als Beispiele nennen wir

$$f(x) = x^2 + 1 \stackrel{\text{mod}\,2}{=} x^2 + 2x + 1 = (x+1)(x+2) \tag{2.13}$$

$$f(x) = x^2 + x + 1 \tag{2.14}$$

Das erste dieser Polynome lässt sich durch Addieren des Ausdrucks $2x$, der unabhängig vom Wert x modulo 2 immer 0 ist, auf eine der binomischen Formel zurückführen. Es ist daher nicht primitiv. Im Gegensatz dazu ist das zweite Polynom (2.14) primitiv. Man vergleiche diese Situation mit Polynomen über $\mathbb{C}$, die immer in Linearfaktoren zerfallen. Eine entsprechende Aussage gilt hier nicht!

Auch das Polynom (2.12) ist primitiv. Weitere Beispiele für primitive Polynome relative kleiner maximaler Ordnung findet man in *Numerical Recipes*.

Eine besondere Rolle unter den primitiven Polynomen spielen die Zierler-Trinome [Zierler, (1969)], die sich in der Form

$$f(x) = 1 + x^a + x^b, \tag{2.15}$$

mit $(b > a)$ schreiben lassen. Sie ergeben für große b sehr lange nichtwiederkehrende Bitsequenzen und führen auch noch zu besonders einfachen Rekursionsvorschriften. Das Paar $(a,b) = (4187, 9689)$ wird häufig in Zufallszahlengeneratoren verwendet. Eine weitere populäre Kombination ist $(103, 250)$. Diese Komination ist zwar nicht prim, ihr werden aber von Kirkpatrick und Stoll (1982) gute statistische Eigenschaften bescheinigt. Das größte bis heute veröffentlichte ist $(54454, 132049)$ (Herringa 1992).

2.3.2 Korrelation in Lagged-Fibonacci-Generatoren

Ein von Compagner (1992) aufgestelltes Theorem besagt, dass bei jeder so erzeugten Bitfolge, deren Rekursionsvorschrift auf ein primitives Polynom zurückgeht, alle Korrelationen verschwinden, die kürzer sind als die Polynomordnung, d.h.

$$\langle x_i x_{i+n} \rangle - \langle x_i \rangle^2 = 0 \tag{2.16}$$

für alle $n < b$. Der Mittelwert ist dabei über *alle* Bits der Folge zu erstrecken.

Man beachte jedoch die neuere Literatur (Ziff, 1998) zu Korrelationen in *kurzen* Sequenzen der Zierler-Generatoren. Ziff schlägt einige Polynome mit 5 Summanden vor und testet diese an speziellen Modellen.

2.3.3 Implementierung

Zur Umwandlung der erzeugten binären Sequenzen in ganze Zahlen (etwa 32 Bit `unsigned` Variablen) bieten sich zwei Möglichkeiten an:

1. Man läßt 32 Fibonacci-Generatoren parallel laufen. Wir werden unten sehen, wie das sehr effizient duchgeführt werden kann. Das Problem hierbei ist die Initialisierung, da die 32 Startsequenzen nicht nur jede für sich, sondern auch untereinander unkorreliert sein müssen. Ihre Qualität beinflusst eintscheidend die Qualität der produzierten Zufallszahlen

2. Man nimmt ein 32 Bit langes Stück aus einer Sequenz heraus. Das Problem dieses Verfahrens ist, dass es relativ langsam ist, denn für jede Zufallszahl müssen 32 neue Folgenglieder der binären Sequenz erzeugt werden. Weiterhin weisen so erzeugte Zufallszahlen nach Knuth (Seminumerical Algorithms, 2. Auflage, Addison-Wesley) starke Korrelationen auf.

Wir besprechen die Implementierung des Kirkpatrick-Stoll Generators. Dazu wird ein Feld von `unsigned` Variablen angelegt, in dem so viele der bereits erzeugten Bits gespeichert werden können, wie zur Auswertung der Rekursionsvorschrift, hier

$$x_0 = x_{103} + x_{250} \tag{2.17}$$

nötig sind.

```
// Feld speichert die 250 letztgenerierten Bits
//  x_1 in pos 0, x_103 in pos 102, x_250 in pos 249
static unsigned  rnd[250];
// Indizes um 1 hoeher als Positionen im C-Feld, wird
//  im ersten Aufruf korrigiert.
static int       idx0 = 250;
static int       idx1 = 103;
```

```
unsigned kps_generator(){
   // Indizes um 1 verschieben, so dass auf die
   //  naechstjuengeren Bits verwiesen wird.
   // wichtig ist, dass idx1 nach 103 Schritten
   //  das jeweils neu generierte bit erreicht
   // wir beachten, dass negative Zahlen am oberen
   //  Feldrand wieder eingeblendet werden muessen
   if ( --idx0 < 0 ) idx0 = 249;
   if ( --idx1 < 0 ) idx1 = 249;

   // aeltestes bit (idx0) mit neuem ueberschreiben
   //  und das Ergebnis als neue Zufallszahl zurückgeben
   return ( rng[idx0] = rng[idx0]^rng[idx1] );
};
```

Zur Bestimmung der Anfangswerte kann man z.B. einen kongruentiellen Generator verwenden—möglichst mit Bays-Durham Mischung. Mit diesem läuft man dann bitweise durch alle 32×250 Bits des Feldes `rnd` setzt alle Bits mit Wahrscheinlichkeit 0.5, etwa indem man prüft, ob der Generator eine Zahl geliefert, die größer als die Hälfte des Maximalwertes ist. Damit umgeht die Probleme mit Korrelationen in den niedrigstwertigen Bits von kongruentiellen Generatoren.

2.4 Mögliche Tests für Zufallszahlengeneratoren

1. Verteilung nachprüfen.
2. Der Mittelwert aller ZZ sollte 0.5 betragen.
3. Der Mittelwert der einzelnen Bits muss 0.5 sein. D.h., es müssen gleichviele Nullen wie Einsen generiert werden.
4. n-tupel $(x_i, x_{i+1}, \ldots, x_{i+n-1})$ in kartesischen Koordinaten auftragen (n-cube-test) und auf homogene Besetzung prüfen. Man untersucht damit die n-Punkte-Korrelationen auf kurze Distanzen.
5. Korrelationsfunktionen (siehe Kapitel 1) müssen verschwinden.
6. Spektraltest, d.h. Leistungsspektrum bestimmen. Man kann damit die Korrelation auf großen Distanzen prüfen. Gute Zufallszahlen liefern als Spektrum weißes Rauschen.
7. Prüfen, daß Teilsummen der Sequenz normalverteilt sind (z.B. unter Benutzung eines χ^2-Tests, Siehe N.J. Giordano, “Computational Physics”, Prentice Hall 1997,S. 164).

2.5 Erzeugung beliebig verteilter ZZ

Die oben beschriebenen Generatoren erzeugen gleichförmig verteilte Zufallszahlen. Es sollen nun zwei Möglichkeiten erläutert werden, die es erlauben, aus solchen gleichförmig verteilten Zufallszahlen x neue y mit einer anderen, weitgehende beliebigen Verteilung $n(y)$ zu erzeugen.

2.5.1 Verwerfungsmethode

Die einfachste Methode zur Erzeugung beliebiger Verteilungen ist die sogenannte *Verwerfungsmethode*. Falls $n(y)$ überall kleiner als A ist und $y \in [0,B]$ liegt, dann kann man $n(y)$ durch ein Rechteck mit den Seitenlängen A und B umschließen.

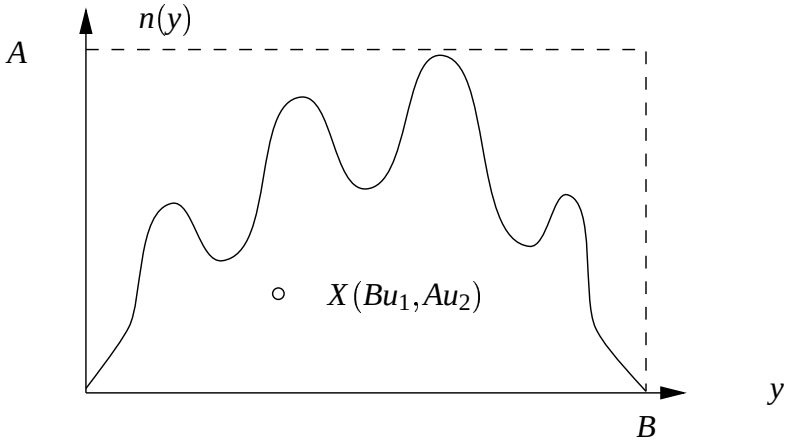


Abb. 2.7 *Verwerfungsmethode*

Wir nehmen an, dass unser gleichförmiger Generator, etwa durch geeignete Skalierung, Zufallszahlen u_i aus dem Intervall $[0,1]$ erzeugt. Wir berechnen damit jetzt ein Paar (Bu_1, Au_2) innerhalb des Rechtecks. Gilt nun $n(Bu_1) < Au_2$, dann verwirft man das Paar und würfelt ein neues. Im umgekehrten Fall wird $y_i = Bu_1$ als Zufallszahl aufgefasst. Diese realisiert die gewünschte Verteilung $n(y)$.

Damit nicht zuviele Zufallszahlen verworfen werden müssen, sollte das Rechteck die Verteilung möglichst eng umschließen. Gegebenenfalls kann man die $n(y)$ auch durch nicht-gleichförmige Verteilung umschließen, die mit der Methode im folgenden Abschnitt erzeugt werden kann. Weitere Details finden sich in den *Numerical Recipes*).

2.5.2 Transformationsmethode

Wir wollen eine Funktion f suchen, die angewandt auf die im Intervall $[0,1]$ gleichförmig verteilten Zufallszahlen neue Zufallszahlen $v_i = f(u_i)$ erzeugt, die einer vorgegebenen Verteilung genügen.

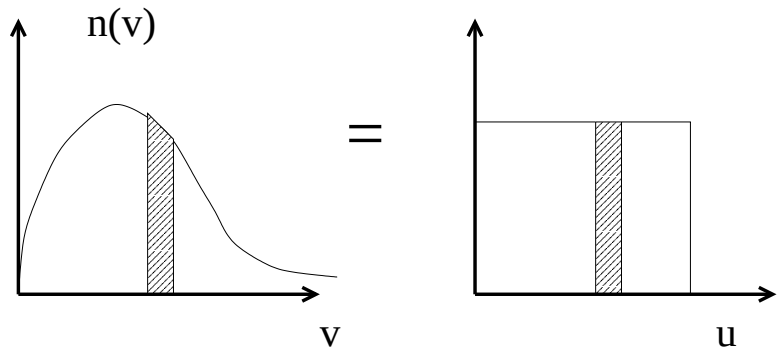


Abbildung 2.2: Ungleichverteilungen werden auf Gleichverteilungen mit derselben Normierung abgebildet

Betrachten wir ein Intervall um u der Breite du. Dieses wird abgebildet auf eines um $v = f(u)$ der Breite $|f'(u)|du$, wenn f differenzierbar ist. Damit $n(v)$ auf 1 normiert bleibt, muss der Flächeninhalt unter den jeweiligen Verteilungen gleich sein, also

$$du = n(v)|f'(u)|du = n(v)|dv/du|du, \tag{2.18}$$

daher $n(v)|dv/du| = 1$. Nehmen wir an, dass wir eine Funktion mit positiver Ableitung dv/du suchen, dann ergibt Multiplikation mit du und Integration ab 0

$$\int_0^v n(v')dv' = u. \tag{2.19}$$

Bezeichnen wir jetzt mit $N(v)$ eine Stammfunktion von $n(v)$, dann erhalten wir durch Auflösung der obigen Gleichung nach v :

$$N(v) - N(0) = u \tag{2.20}$$
$$v = f(u) = N^{-1}(u + N(0)) \tag{2.21}$$

Dies ist die gesuchte Transformationsfunktion.

Beispiel: Poissonverteilung

Als Beispiel betrachten wir die Poissonverteilung. Sei also $n(v) = \exp(-v)$. Damit ist $N(v) = -\exp(-v)$, $N(0) = -1$, also

$$\begin{aligned} -\exp(-v) + 1 &= u \\ \exp(-v) &= 1 - u \\ v &= -ln(1-u) \end{aligned}$$

Tatsächlich ist die Ableitung dieser Funktion nach u positiv im Intervall $[0,1)$, wie wir das bei der Herleitung vorausgesetzt haben.

Gaußverteilte Zufallszahlen

Möchte man Zufallszahlen mit einer Verteilung $n(v) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{v^2}{2})$, d.h. einer Normalverteilung mit Standardabweichung $\sigma = 1$, erhalten, dann funktioniert das obige Verfahren nicht mehr so einfach, da sich das Integral über die Gaußfunktion nicht analytisch angeben lässt und wir somit auch nicht die Umkehrfunktion in analytischer Form angeben können.

Wir müssen einen Trick anwenden, der letztlich eine Verallgemeinerung der Formel (2.18) auf mehr als eine Dimension darstellt. Die Ableitung der Funktion f wird dabei ersetzt durch die Funktionaldeterminante einer vektorwertigen Transformationsfunktion $v_i(u_i \dots)$

$$n(v_1, v_2, \dots) \det(\frac{\partial(v_1, v_2, \dots)}{\partial(u_1, u_2, \dots)}) = 1. \tag{2.22}$$

Hier ist vorausgesetzt, dass $(u_1, u_2 \dots)$ im n-dimensionalen Einheitwürfel gleichverteilt ist.

Da wir in einer Dimension keine Transformationsfunktion finden können um daraus gleichverteilte gaußverteilte Zufallszahlen zu machen, versuchen wir unser Glück in 2 Dimensionen. Wir suchen $v_1(u_1, u_2)$ und $v_2(u_1, u_2)$ so dass

$$\det\left(\frac{\partial(u_1, u_2, \dots)}{\partial(v_1, v_2, \dots)}\right) = \frac{1}{2\pi} \exp(\frac{v_1^2 + v_2^2}{2}). \tag{2.23}$$

Dieser Ausdruck ist das Produkt zweier unabhängiger Gaußfunktionen. Wir wissen, daß sich das Produkt zweier Gaußintegrale in 2 Dimensionen durch eine Integration in Polarkoordinaten $r^2 = v_1^2 + v_2^2$, $\tan\phi = \frac{v_2}{v_1}$ schreiben lässt. Wir berechnen das Integral über einen Kreissektor K des Radius r und Scheitelwinkel ϕ,

$$\begin{aligned}
\int\int_K \frac{1}{\sqrt{2\pi}} \exp(-\frac{v_1^2}{2}) \quad \cdot \quad & \frac{1}{\sqrt{2\pi}} \exp(-\frac{v_2^2}{2}) dv_1 dv_2 && (2.24)\\
= \quad & \int_K \frac{1}{2\pi} \exp(-\frac{v_1^2 + v_2^2}{2}) dv_1 dv_2 && (2.25)\\
= \quad & \frac{1}{2\pi} \int_0^\phi \int_0^r \exp(-\frac{r^2}{2}) r dr d\phi && (2.26)\\
= \quad & \frac{\phi}{2\pi}(1 - \exp(-\frac{r^2}{2})) && (2.27)\\
= \quad & \left[\frac{1}{2\pi} \arctan(\frac{v_2}{v_1})\right] \left[1 - \exp(-\frac{v_1^2 + v_2^2}{2})\right] && (2.28)
\end{aligned}$$

Die beiden Faktoren in eckigen Klammern nehmen Werte zwischen 0 und 1 an. Wir versuchen, sie mit der Umkehrfunktion der Transformationsfunktion $u_1(v_1, v_2)$, $u_2(v_1, v_2)$ zu identifizieren (diese muss ja auch in das Intervall $0 \dots 1$ abbilden) und berechnen

deren Funktionaldeterminate. Es ist nicht zu sehr verwunderlich, dass sich diese als Umkehrung der Integration wieder als das Produkt der Gaussfunktionen ergibt:

$$\det\left(\frac{\partial(u_1,u_2,\dots)}{\partial(v_1,v_2,\dots)}\right) = -\left[\frac{1}{\sqrt{2\pi}}\exp(-\frac{v_1^2}{2})\right]\left[\frac{1}{\sqrt{2\pi}}\exp(-\frac{v_2^2}{2})\right]. \tag{2.29}$$

Da dies (2.23) entspricht, sind wir bis auf die Umkehrung des Systems $u_1(v_1,v_2)$, $u_2(v_1,v_2)$ fertig. Man findet

$$v_1 = \sqrt{-2\ln(1-u_2)}\cos(2\pi u_1) \tag{2.30}$$
$$v_2 = \sqrt{-2\ln(1-u_2)}\sin(2\pi u_1) \tag{2.31}$$

Hier erhält man also jedesmal zwei gaußverteilte Zufallszahlen v_1 und v_2 aus zwei gleichförmig in $[0,1)$ verteilten Zahlen u_1 und u_2. Bei der Implementierung dieser Transformationsfunktion kann man sich noch die Auswertung der trigonometrischen Funktionen ersparen, indem man einen zufälligen Punkt $(\tilde{u}_1,\tilde{u}_2)$ in Einheitskreis ermittelt—durch Verwerfen der außerhalb liegenden Paare. $\tilde{u}_1/\sqrt{\tilde{u}_1^2+\tilde{u}_2^2}$ kann dann als Kosinus eines zufälligen Winkels $2\pi u_1$ und $\tilde{u}_2/\sqrt{\tilde{u}_1^2+\tilde{u}_2^2}$ als Sinus des gleichen Winkels Verwendung finden. Der Radius $u_1=\sqrt{\tilde{u}_1^2+\tilde{u}_2^2}$ tritt an die Stelle der zweiten gleichförmig verteilten Variablen.

Um Zufallszahlen einer beliebigen Standardabweichung σ und mit einem beliebigen Mittelwertes $\bar{v}$ zu erzeugen, multipliziert man v_1,v_2 einfach mit σ und addiert $\bar{v}$.

Kapitel 3

Random Walk

3.1 Brownsche Bewegung

Ein klassisches Beispiel für einen Random Walk liefert die von Robert Brown (1829) untersuchte Bewegung von Blütenstaub in Flüssigkeiten. In Ref [2] findet man sogar noch einen Hinweis auf eine frühere Beobachtung durch den Holländer Ingenhousz (1785). Natürlich unterliegt die Bewegung der Blütenstaubteilchen den Newtonschen Gesetzen und ist auf einer mikroskopischen Skala zwar recht kompliziert, aber differenzierbar und stetig. Die (scheinbare) Unregelmäßigkeit der Bewegung hat ihre Ursache in der großen Beobachtungszeitskala. Wir unterscheiden also:

- Eine makroskopische Zeitskala, in der die Beobachtung der Teilchen stattfindet.
- Eine mikroskopische Zeitskala, die durch die Wechselwirkungszeit der Staubteilchen mit den Flüssigkeitsmolekülen vorgegeben wird.

Die makroskopische Bewegung ist nicht differenzierbar und die Geschwindigkeit ist nicht stetig. Damit können wir keinen Geschwindigkeitsvektor definieren. Wir beobachten vielmehr nur eine Abfolge von Positionen $\vec{x}(t_n)$ mit

$$\vec{x}(t_n) - \vec{x}(t_{n-1}) = \Delta\vec{x}$$

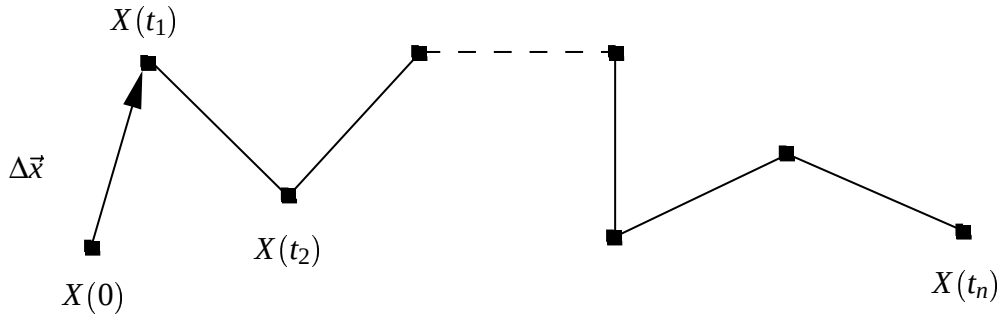


Abb. 3.1 *Random Walk*

$\Delta\vec{x}$ ist eine vektorwertige Zufallsvariable. Trotz der zugrundeliegenden deterministischen Bewegung ist die Summe der vielen Stöße durch Flüssigkeitsteilchen also ei-

ne Zufallsvariable. Die Eigenschaften solcher Bewegungen sollen in dieser Vorlesung näher untersucht werden.

Wir definieren als Random Walk die Folge der (Partial)Summen $\vec{x}_n$ von jeweils n vektorwertigen Zufallsvariablen. Die Eigenschaften des Random Walk werden damit durch die Eigenschaften dieser Zufallszahlen festgelegt und wir werden uns damit jetzt genauer auseinandersetzen.

Einen einfachen Random Walk könnte man z.B. realisieren, indem man Zufallszahlen in Bereich $z_i \in [0,1)$ erzeugt und folgende Zuordnung trifft:

1. Für $z_i \in [0, 0.25)$ rücke um $\Delta\vec{x} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ vor.
2. Für $z_i \in [0.25, 0.5)$ rücke um $\Delta\vec{x} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$ vor.
3. Für $z_i \in [0.5, 0.75)$ rücke um $\Delta\vec{x} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ vor.
4. Für $z_i \in [0.75, 1.0)$ rücke um $\Delta\vec{x} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$ vor.

Im folgenden soll ein Random Walk in einer Dimension betrachtet werden. Die "vektorwertige" Zufallsvariable soll in diesem Fall skalar sein und lediglich die Werte $z_i = \pm 1$ annehmen. Die Wahrscheinlichkeit für $z_i = +1$ betrage q, die Wahrscheinlichkeit für $z_i = -1$ betrage $1-q$. Die im folgenden abgeleiteten Eigenschaften der Random Walks gelten jedoch auch für den allgmeinen, mehrdimensionalen Fall.

Wir wollen zunächst den Erwartungswert der x_n untersuchen:

$$x_n = \sum_{i=1}^{n} z_i$$

Wir identifizieren zunächst den Erwartungswert einer Zufallsvariablen mit eckigen Klammern und bemerken, daß es sich bei der Bildung des Erwartungswertes um eine lineare Operation handelt, wir dürfen also mit Summen vertauschen und Konstanten herausziehen. Mit anderen Worten:

$$EW(x_n) \equiv < x_n > = < \sum_{i=1}^{n} z_i > = \sum_{i=1}^{n} < z_i >$$

Mit der Definition des Erwartungswerts aus Kapitel 1 erhalten wir:

$$\begin{aligned} \sum_{i=1}^{n} < z_i > &= n < z_i > \\ &= n \int [p(z) z] dz \\ &= n \int [q\delta(z-1) + (1-q)\delta(z+1)] z dz \\ &= n(2q-1). \end{aligned}$$

Hier haben wir die Dirac-δ Funktion benutzt, um die diskrete Verteilung unserer Zufallszahlen z_i zu charakterisieren. D.h., wir erhalten als Ergebnis, daß sich unserer

"Walker" im Durchschnitt (= Ensemblemittel über viele Realisierungen) mit der "Geschwindigkeit" $(2q-1)$ vom Ursprung entfernt.

Wir wollen jetzt die Varianz errechnen und notieren zunächst die folgende Identität (die wiederum direkt aus der Linearität der Erwartungswertbildung folgt):

$$< (x_n - < x_n >)^2 > = < x_n^2 > - < x_n >^2 \tag{3.1.1}$$

Während wir oben den Erwartungswert der x_n bereits ermitteln haben, fehlt uns noch das *mittlere Verschiebungsquadrat*

$$\begin{aligned} < {x_n}^2 > &= < \left(\sum_{i=1}^{n} z_i \right)^2 > \\ &= < \sum_{i=1}^{n} {z_i}^2 + \sum_{i \neq j}^{n} z_i z_j > \\ &= \sum_{i=1}^{n} < z_i^2 > + \sum_{i \neq j}^{n} < z_i z_j > \end{aligned}$$

Sind die Zufallszahlen unkorreliert (was wir hier annehmen wollen), dann kann man im hinteren Term schreiben $< z_i z_j > = < z_i >< z_j >$, und da beide Erwartungswerte natürlich gleich sind $< z_i >< z_j > = < z_i >^2$. Eingesetzt ergibt sich damit

$$< {x_n}^2 > = n < {z_i}^2 > + n(n-1) < z_i >^2$$

Den Erwartungswert von $< z_i > = (2q-1)$ haben wir bereits berechnet, der Erwartungswert von $< {z_i}^2 >$ beträgt

$$< {z_i}^2 > = (+1)^2 q + (-1)^2 (1-q) = 1$$

Trägt man nun alle diese Einzelergebnisse zusammen, erhält man schließlich für die Varianz:

$$\begin{aligned} < (x_n - < x_n >)^2 > &= < {x_n}^2 > - < x_n >^2 \\ &= n < z_i^2 > + n(n-1) < z_i >^2 - n^2 < z_i >^2 \\ &= n < z_i^2 > - n < z_i >^2 \\ &= n - n(2q-1)^2 \\ &\sim n. \end{aligned}$$

Hier haben wir das Zeichen $\sim$ eingeführt, welches die Bedeutung von "asymptotisch proportional zu" hat (asymptotisch heißt hier für große Werte von n).

Im Falle, daß $q = \frac{1}{2}$ ist, kann man sich in 2 Dimensionen etwa vorstellen, daß der Random Walker eine Kreisfläche erforscht, deren Radius sich wie die Wurzel aus der Varianz vergrößert. Falls q ein wenig verschieden von $\frac{1}{2}$ ist, dann verschiebt sich dieser Kreis auch noch während er wächst und überdeckt einen sich langsam vergrößernden Schlauch.

Die Abhängigkeit der eben berechneten Größen von n gilt in allen Dimensionen und sei hier noch einmal im Überblick zusammengefaßt:

q	$< x_n >$	$< {x_n}^2 >$	$< (x_n - < x_n >)^2 >$
0.5	0	$\sim n$	$\sim n$
$\neq 0.5$	$\sim n$	$\sim n^2$	$\sim n$

Für $q = \frac{1}{2}$, d.h. für den Fall, daß $z_i = +1$ und $z_i = -1$ gleich wahrscheinlich sind, bleibt der Random Walker im Mittel immer an einer Stelle – wie man es anschaulich auch erwartet. In allen anderen Fällen wandert er (im Mittel) proportional zu n von seinem Ausgangspunkt weg. Das deckt sich mit der Aussage des zentralen Grenzwertsatzes, welcher folgendes besagt:

Hat z eine (ansonsten beliebige!) Verteilung $p(z)$ so, daß das Integral $\int p(z) z^2 dz$ konvergiert, dann geht die Summe $\sum^n z_i$ gegen eine Gaußverteilung mit Mittelwert $\sim n < z_i >$ und Varianz $\sim n < {z_i}^2 >$.

Eine genauere und schärfere Fassung des zentralen Grenzwertsatzes läßt sich unter Benutzung des Faltungstheorems herleiten und ist z.B. in Reif [5] nachzulesen.

Der Random Walk spielt eine große Rolle in vielen Bereich der Physik. Er ist nicht nur als Beschreibung der Brown'schen Bewegeung nützlich. Z.B. läßt sich auch die geometrische Konformation einer langen Polymerkette als Random Walk auffassen, die einzelnen Monomere sind die Schritte Δx und deren Anzahl N ist ein Maß für die Polymermasse und gleichzeitig für die Länge des für die Beschreibung nötigen Random Walks. Polymere sind auch ein Beispiel dafür, daß die einzelnen Schritte nicht unkorreliert sein müssen, sondern daß z.B. die Ausrichtung eines Monomers sehr wohl von der Ausrichtung des Monomers "davor" abhängen kann. Erst nach einigen Monomeren (Korrelationslänge, vergleiche mit dem Begriff der in der ersten Vorlesung eingeführten Korrelationszeit) "vergißt" das Polymer die Ausrichtung.

Ein weiteres Beispiel für einen eindimensionalen Random Walk ist die Kursentwicklung an der Börse. Der mittlere Aktienkurs ist täglichen Schwankungen unterworfen, die als Schritte in einem Random Walk Prozeß aufgefaßt werden können. Allerdings zeigt sich bei einer genauen Analyse, daß die Varianz der Schritte nicht endlich ist. Damit gelten die oben in der Tabelle zusammengefaßten Resultate für normale Random Walks, deren Schritte endliche Varianz haben, nicht mehr. Dies ist einer der Gründe, warum die Börse so interessant (und ruinös) sein kann.

Wir wollen z.B. ein Polymer charakterisieren. Wir können das dadurch erreichen, daß wir seine Gesamtausdehnung angeben, den maximalen Abstand $r_{\max}$ zwischen zweien seiner Monomere. Da ein einzelnes Polymer sehr starken Fluktuationen seiner Form unterworfen ist, ist so eine Größe, die nur von der Lage zweier Monomere abhängt, statistisch keine "gute" Variable, wenn man an mittleren Eigenschaften interessiert ist. Besser sind Größen, in die alle Monomere eingehen.

Eine sehr gut geeignete solche Größe ist uns bereits aus den einführenden Physikvorlesungen bekannt: der Trägheitsradius. Es galt für das Trägheitsmoment θ eines aus N gleichschweren Teilchen der Masse m zusammengesetzten Körpers:

$$\theta = m \sum^N {r_\perp}^2,$$

$r_\perp$ ist der Abstand zur Drehachse des Körpers (wir denken uns eine solche durch den Schwerpunkt verlaufend). Wir definieren nun den Trägheitsradius R_g (radius of gyration) durch

$$\theta = mNR_g^2.$$

Gleichsetzen liefert

$$R_g^2 = \frac{1}{N}\sum^N {r_\perp}^2 = < {r_\perp}^2 > .$$

In unserer Terminologie ist $< {r_\perp}^2 >$ nichts weiter als das mittlere Verschiebungsquadrat unseres Random Walks, wenn wir x_n vom Massenschwerpunkt aus rechnen. Allgemein benutzen wir:

$$R_g^2(n) \equiv \frac{1}{N}\sum^N x_i^2 - \left(\frac{1}{N}\sum^N x_i\right)^2 .$$

3.2 Diffusion

Diffusion, d.h. der Ausgleich lokaler Konzentrationunterschiede, wird beschrieben durch die (Diffusions-)Gleichung:

$$\frac{\partial c}{\partial t} = D\Delta c(\vec{x},t)$$

Δ ist der Laplaceoperator, $c(\vec{x},t)$ sei die Konzentration, für die Normierung

$$\int c(\vec{x},t)dx^3 = 1$$

gelten soll. Wir wollen zunächst darauf hinweisen, daß die Konzentration als ein Ensemblemittel über Random Walks verstanden werden kann. Dazu wird der Raum in kleine Zellen eingeteilt, und in jedem Moment die Anzahl der gerade in diesen Zellen befindlichen Walker gezählt und durch die Gesamtzahl der Walker und das Zellvolumen dividiert. Damit ergibt sich automatisch die obige Normierung, die nichts anderes besagt, als daß ein Walker sich zur Zeit t irgendwo im Raum befinden muß.

Wie beim Verschiebungsquadrat des Random Walks soll auch hier das zweite "Moment" des Ortes berechnet werden:

$$< \vec{x}^2(t) > = \int \vec{x}^2 c(\vec{x},t)dx^3$$

Mit der Diffusionsgleichung erhält man

$$\begin{aligned}\frac{\partial}{\partial t}\int \vec{x}^2 c(\vec{x},t)dx^3 &= D\int \vec{x}^2\Delta c(\vec{x},t)dx^3 \\ &= D\int \nabla(\vec{x}^2\nabla c(\vec{x},t))dx^3 - D\int \nabla\vec{x}^2\nabla c(\vec{x},t)dx^3\end{aligned}$$

Das vordere Integral in der letzten Zeile wandelt man mit Hilfe des Satzes von Gauß in ein Oberflächenintegral um, welches mit obiger Normierung von c verschwindet (Aufgrund der Normierung muß c und Δc im Unendlichen gegen 0 gehen). Das zweite Integral wird erneut partiell integriert, wobei wieder das entstehende Oberflächenintegral verschwindet. So erhält man schließlich:

$$\frac{\partial}{\partial t}\int \vec{x}^2 c(\vec{x},t)dx^3 = D\int \Delta\vec{x}^2 c(\vec{x},t)dx^3$$

Mit $\Delta\vec{x}^2 = 2d$, wobei d die Dimension von $\vec{x}$ ist, und der obigen Normierung kann man schreiben:

$$\frac{\partial}{\partial t} < \vec{x}^2(t) >= 2dD,$$

und integriert

$$< \vec{x}^2(t) >= 2dDt.$$

Wir identifizieren jetzt die Zeit t im Diffusionsproblem mit der Länge n des Random Walk und vergleichen die vorige Beziehung mit unserer Formel für den symmetrischen Random Walk:

$$\begin{aligned} < x^2(t) > &= 2dDt \\ < x_n^2 > &= n < z_i > . \end{aligned}$$

Damit müssen, damit wir einen Diffusionprozess mit Diffusionskonstante D nachbilden können, die Schrittweiten $\Delta x = z_i$ die Bedingung

$$< z_i >= 2dD$$

erfüllen. Diese Möglichkeit, einen Diffusionprozeß durch Random Walker nachbilden zu können, bietet für zahlreiche physikalische Prozesse Vorteile: so kann beispielsweise eine chemische Reaktion auf der Basis der Reaktanden und anschliessender Diffusion der entstandenen Spezies nachgebildet werden. Häufig erschließen sich so Erkenntnisse, die man durch “einfache” Kontinuumsgleichungen wie etwa die Diffusionsgleichung nicht erhalten kann [4].

Kapitel 4

Perkolation

4.1 Definition der Perkolation, Gitter

Als *Perkolation* bezeichnet man eine Klasse von Modellen, die dadurch gekennzeichnet sind, daß Teilchen zufällig und unabhängig voneinander auf die Plätze eines Gitters gesetzt werden.

Aus computertechnischen Gründen arbeitet man am häufigsten auf Quadratgittern bzw. im Dreidimensionalen auf kubischen Gittern.

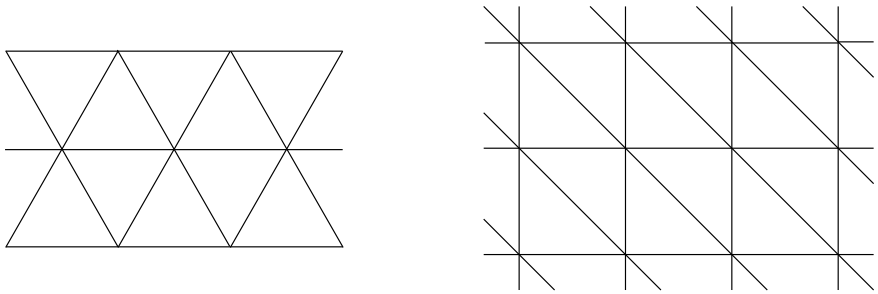


Abb. 4.1 *links: Ein Dreicksgitter rechts: Verbindungen in der Computerimplementation*

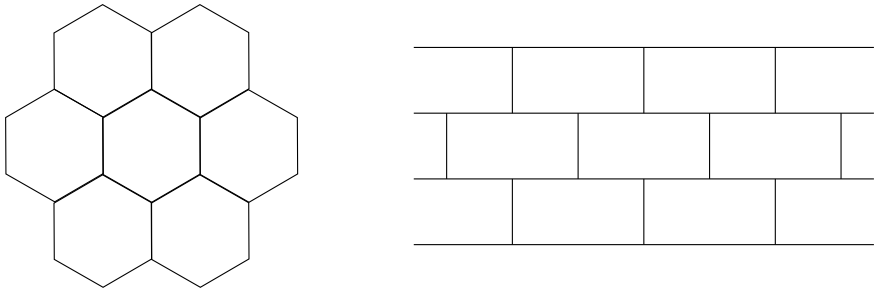


Abb. 4.2 *links: Ein Bienenwabengitter rechts: Die Implementierung auf dem Computer*

Man kann nun zwei Arten der Perkolation unterscheiden:

1.) *site-Perkolation* – Man besetzt die Plätze eines Gitters mit der Wahrscheinlichkeit p. Wichtig ist, daß die Besetzung jedes Platzes unabhängig von der Besetzung der anderen Plätze erfolgt. Um dies zu erreichen, wählt man zu jedem Gitterplatz eine homogen in [0,1] verteilte ZZ x. Ist $x < p$, so wird der Platz besetzt, sonst bleibt er frei.

2.) *bond-Perkolation* – Hier werden nicht die einzelnen Gitterplätze besetzt, sondern es werden zufällig die Verbindungen zwischen benachbarten Gitterplätzen belegt.

4.2 Randbedingungen

Für Gitter- und gleichermaßen für Kontinuumsmodelle ist die Festlegung von Randbedingungen (RB) wichtig. Typischerweise unterscheidet man:

1. freie RB – Ein Gitterplatz (oder eine -bindung) am Rand ist nur mit Nachbarn im Inneren des Gebietes verbunden und sein Zustand unterliegt keinerlei Einschränkungen. In Fall der Straße aus Kapitel 1 könnten freie RB bedeuten, dass Autos mit einer bestimmten Rate an den Streckeanfang gebracht werden, die Strecke befahren und dann jedes Auto zu dem Zeitpunkt, zu dem es das Streckenende erreicht, dort entfernt wird. Bei kontinuierlichen Gleichungen legt man an den Rändern die Ableitungen fest (Neumann RB).

2. feste RB – Im Perkolationsmodell entspräche das einem festen Zustand der Randplätze, etwa einer Besetzung aller Randplätze oder links besetzten und rechts unbesetzten Randplätzen. Die Kontinuumsentsprechung ist ein am Rand eingeprägter fester Randwert (Dirichlet RB).

3. periodische RB – In diesem Fall verwendet man als Zustände der Randplätze einfach die der jeweils gegenüberliegenden Systemoberfläche. Dies entspricht einer periodischen Wiederholung des Systems im ganzen Raum. Periodische Randbedingungen legen zwar das System nicht so fest wie feste oder freie RB, können jedoch bei Wellenphänomenen die Systemgröße fälschlicherweise als charakteristische Länge auszeichnen.

   Im Falle der Straße bedeuten peridische RB, dass Autos, die die Strasse verlassen, mit gleicher Geschwindigkeit am anderen Ende wieder auftauchen.

4. antiperiodische RB – Die Zustände am Rand entsprechen gerade den umgekehrten auf der Gegenseite.

5. helikale RB – werden gerne in höheren Dimensionen aus programmiertechnischen Gründen verwendet. Etwa sind zwei- und mehrdimensionale Felder in Programmiersprachen durch eine eindeutige Abbildung der Indizes auf einen eindimensionalen Datenbereich implementiert. Schritte nach links, rechts, oben,

unten, etc. entsprechen bestimmten festen Indexinkrementen oder -dekrementen im zugrundeliegenden eindimensionalen Speicherbereich. Statt das hochdimensionale Feld zu manipulieren, kann man alternativ einfach mit diesen Inkrementen auf der eindimensionalen Entsprechung arbeiten. Verwendet man jetzt eine *modulo* Operation, um sicherzustellen, dass man niemals aus dem zulässigen Indexbereich herausläuft, so erhält man "fast" periodische Ränder, die allerdings mit einer Verschiebung um einen Gitterplatz einhergehen.

4.3 Cluster

4.3.1 Perkolierendes Cluster

Die zentrale Größe im Zusammenhang mit der Perkolation sind *Cluster* verbundener Teilchen. (Bei der site-Perkolation sind zwei benachbarte Plätze dann verbunden, wenn sie beide besetzt sind.) Als Cluster definiert man alle Gitterplätze, die durch einen Pfad besetzter Verbindungen zusammenhängen. Man sagt, daß ein System perkoliert (durchsickert), wenn ein Cluster zusammenhängender Teilchen existiert, das zwei gegenüberliegende Systemränder verbindet oder (im Falle unendlicher Systeme) eben unendliche Ausdehnung hat.

Ob es ein perkolierendes Cluster gibt, prüft man mit dem Algorithmus des "Verbrennens":

1. Es sei ein Gitter N(x,y) gegeben. Ist der Gitterplatz (x,y) mit einem Teilchen besetzt, habe N(x,y) den Wert 1, sonst ist N(x,y)=0. Die Gitterseite, auf der man startet soll den y-Wert 0 haben, die gegenüberliegende Seite den Wert L.
2. Zur t=0 gibt man nun allen besetzten Gitterpunkten mit y=0 den Wert 2 (oder sonst eine Zahl $\neq 0, 1$), d.h. man "verbrennt" sie. Gleichzeitig merkt man sich ihre Positionen in einer Liste, sowie ihre Anzahl B in einem Zähler B.
3. Es werden alle Nachbarn, der B im vorhergegangenen Schritt verbrannten Gitterplätze untersucht. Sind sie besetzt und unverbrannt, dann werden sie auch verbrannt. Die Positionen der neu verbrannten Plätze und ihre Anzahl wird wieder gespeichert. t wird um eins erhöht.
4. Der 3. Schritt wird so lange wiederholt, bis ein Platz verbrannt wird, der die y-Koordinate L hat, oder bis B=0 ist. Im ersten Fall perkoliert das Cluster. t gibt dann die Länge des kürzesten Weges an. Im zweiten Fall gibt es kein perkolierendes Cluster.

Über mehrere Konfigurationen gemittelt, kann man so die Perkolationswahrscheinlichkeit in Abhängigkeit von der Besetzungwahrscheinlichkeit eines Gitterpunktes ermitteln. Man findet eine kritische Besetzungwahrscheinlichkeit p_c, nämlich den niedrigsten p-Wert, bei dem ein perkolierender Cluster existiert. Sie ist jedoch abhängig von der Art des Gitters und der Art der Perkolation. Für das Quadratgitter ist $p_c = 0.59275$. Für das Dreiecksgitter beträgt $p_c = 0.5$.

Gittertyp	site	bond
Wabengitter	0,6962	0,65271
Quadratgitter	0,592746	0.50000
Dreiecksgitter	0,500000	0,34729
Diamantgitter	0,43	0,388
einfach kubisches Gitter	0,3116	0,2488
BCC	0,246	0,1803
FCC	0,198	0,119
hyperkubisch (4d)	0,197	0,1601
hyperkubisch (5d)	0,141	0,1182
hyperkubisch (6d)	0,107	0,0942
hyperkubisch (7d)	0,089	0,0787

Kritische Besetzungswahrscheinlichkeit p_c verschiedener Gittertypen bei site- und bond-Perkolation.

4.3.2 Typische Clustergröße

Es soll nun die Wahrscheinlichkeit dafür betrachtet werden, daß zwei Gitterpunkte im Abstand r zum gleichen Cluster gehören. Die so bestimmte Wahrscheinlichkeitsverteilung $g(r)$ ist wieder eine Art Korrelationsfunktion. Sie wird als *Konvektivitätskorrelationsfunktion* bezeichnet. In diesem Fall sind zwei Gitterpunkte korreliert, wenn sie verbunden sind. Zur Bestimmung von $g(r)$ wählt man einen beliebigen besetzten Punkt im Gitter als Ursprung. Im nächsten Schritt markiert man nun alle Punkte, die mit dem Ursprung verbunden sind (z.B. durch den Algorithmus des Verbrennens). Um den Ursprung zieht man nun konzentrische Kreise im Abstand Δr. $g(r)$ ergibt sich dann in zwei Dimensionen als

$$g(r) = \frac{\text{Anzahl besetzter Punkte im Abstand } [r, r+\Delta r]}{2\pi r \Delta r}$$

Für $p \neq p_c$ ist $g(r)$ eine Exponentialfunktion $g \propto exp(-\frac{r}{\xi})$. Die Korrelationslänge ξ beschreibt für $p < p_c$ den typischen Durchmesser endlichen Clusters. Für $p > p_c$ ist ξ die typische Länge eines Clusters ohne das unendliche Cluster oder auch die typische Größe der Löcher im unendlichen Cluster. Im Bereich um p_c gehorcht ξ folgendem Potenzgesetz:

$$\xi \propto |p - p_c|^{-\nu}$$

ν ist eine universelle Konstante. Insbesondere ist ν also unabhängig vom Gitter- oder Perkolationstyp. Man findet:

$$\nu = \begin{cases} 4/3 & \text{in 2 Dimensionen} \\ 0.88 & \text{in 3 Dimensionen} \end{cases}$$

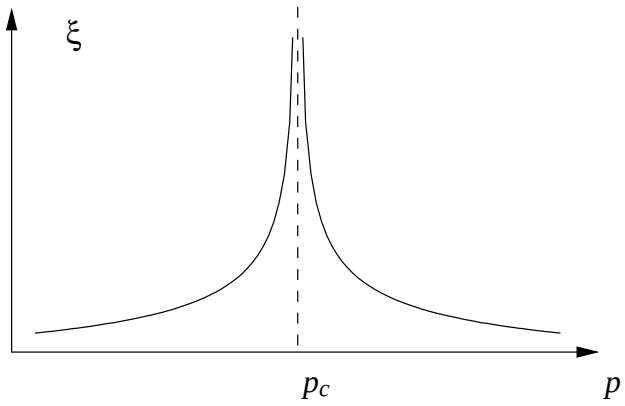


Abb. 4.3 *kritisches Verhalten der charakteristischen Länge*

Eine andere (äquivalente) Definition für ξ ist

$$\xi^2 = \frac{\sum R_s^2 s^2 n_s}{\sum s^2 n_s}$$

s ist die Zahl der besetzten Gitterplätze im Cluster. n_s ist die Zahl der Cluster, die s Gitterplätze enthalten, pro Gitterplatz. $\sum$ läuft über alle s, jedoch ohne die unendlichen Cluster. Der "Clusterradius" R_s berechnet sich nach:

$$R_s^2 = \frac{\sum_{i=1}^{s} s^2 |\vec{r}_i - \vec{r}_0|^2}{\sum_{i=1}^{s} s^2}$$

mit der Schwerpunktkoordinate

$$\vec{r}_0 = \frac{1}{s} \sum_{i=1}^{s} \vec{r}_i$$

wobei die Summe hier über alle Gitterplätze s im Cluster läuft.

4.3.3 Clustergrößenverteilung

Eine zentrale Größe bei der Beschreibung von Clustern ist die bereits oben eingeführte Clustergrößenverteilung n_s. Aus den Momenten von n_s lassen sich alle wesentlichen Größen zur Beschreibung des Systems gewinnen.

0. Moment: $\sum_s' n_s = Z$

Die gestrichene Summe soll nur über die endlichen Cluster laufen. Das 0. Moment liefert die Anzahl der endlichen Cluster pro Gitterplatz.

1. Moment: $\sum_s' s n_s = p$

Das erste Moment liefert für $p < p_c$ genau die Besetzungswahrscheinlichkeit. Für $p > p_c$ liefert die Differenz zur Besetzungswahrscheinlichkeit die Wahrscheinlichkeit P_∞ dafür, daß ein Punkt zum unendlichen Cluster gehört.

$$P_\infty = p - \sum_s' s n_s$$

Auch P_∞ gehorcht in der Nähe von p_c wieder einem Potenzgesetz:

$$P_\infty(p) \propto (p - p_c)^\beta$$

Die universelle Konstante β beträgt diesmal:

$$\beta = \begin{cases} 5/36 & \text{in 2 Dimensionen} \\ 0.41 & \text{in 3 Dimensionen} \end{cases}$$

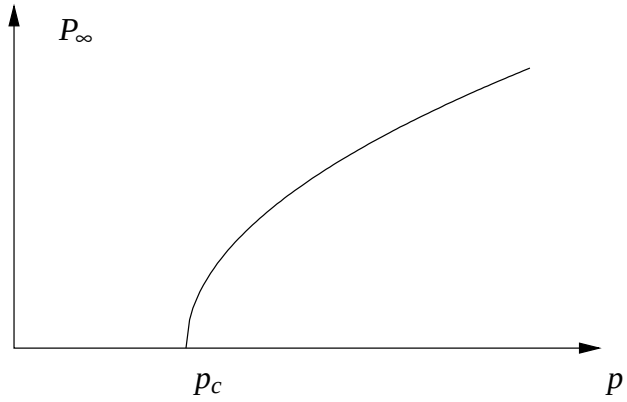


Abb. 4.4 *Verhalten des Ordnungsparameters P_∞ in Abhängigkeit von der Besetzungswahrscheinlichkeit p. p_c ist die kritische Besetzungswahrscheinlichkeit.*

P_∞ ist ein "Ordnungsparamter" der den Übergang zum perkolierenden System beschreibt.

2. Moment: $\sum_s' s^2 n_s = \chi$

χ ist proportional zur mittleren Anzahl von Plätzen in einem endlichen Cluster. In der Nähe von p_c gehorcht χ dem Potenzgesetz

$$\chi \propto (p - p_c)^{-\gamma}$$

Die universelle Konstante γ beträgt diesmal:

$$\gamma = \begin{cases} 43/18 & \text{in 2 Dimensionen} \\ 1.7 & \text{in 3 Dimensionen} \end{cases}$$

Betrachtet man die Analogie zwischen Perkolation und dem Curiepunkt eines Ferromagneten, entspricht p der Temperatur. P_∞ verhält sich dann wie die Magnetisierung und χ wie die Suszeptibilität.

Auf dem Computer bestimmt man $n_s = N_s/N$ mit dem sogenannten *Hoshen-Kopelman* Algorithmus (1976):

1. Es sei das Gitter $N(i,j)$ gegeben mit $N(i,j) = 0$ für alle unbesetzten Plätze und $N(i,j) = 1$ für alle besetzten Plätze. Man führt auch ein Merker-Feld $M(k)$ ein, in welchem die Anzahl aller Gitterplätze mit dem Wert k gespeichert werden.

2. Man startet nun in der ersten Zeile $(i = 1)$ mit dem ersten Gitterpunkt $(j = 1)$. Der Anfangswert für k ist 2. Der erste besetzte Gitterplatz erhält den neuen Wert $N(1, j) = k$. $M(k)$ wird entsprechend auf 1 gesetzt. Ist der Punkt $j+1$ ebenfalls besetzt, bekommt auch er den Wert k zugewiesen und $M(k)$ wird um eins erhöht. Jedesmal, wenn auf einen unbesetzten Platz ein besetzter Platz folgt, wird k um eins erhöht.
3. Findet man in den folgenden Zeilen einen besetzten Platz (i, j), prüft man ob der links benachbarte Platz $(i, j-1)$ oder der darüberliegende Platz $(i-1, j)$ besetzt ist. Dabei können folgende Fälle auftreten:

   a) Keiner der beiden Plätze ist besetzt. Dann erhöht man k um eins und setzt $N(i, j) = k$.

   b) Einer der beiden Plätze ist besetzt und hat den Wert k. Dann setzt man $N(i, j) = k$ und $M(k) = M(k) + 1$.

   c) Beide Plätze sind besetzt und haben den gleichen Wert k. Dann verfährt man gleich wie in b).

   d) Beide Plätze sind besetzt und haben verschiedene k-Werte k_1 und k_2. Man wählt den k-Wert des größeren Clusters (hier: k_1) und macht dann folgende Zuordnungen:

$$\begin{aligned} N(i, j) &= k_1 \\ M(k_1) &= M(k_1) + M(k_2) + 1 \\ M(k_2) &= -k_1 \end{aligned}$$

   Der negative Inhalt von $M(k_2)$ gibt nun nicht mehr die Anzahl der Gitterpunkte an, aus denen das Cluster besteht, sondern ist ein Zeiger auf das "Wurzelcluster".

   e) Einer oder beide Plätze haben bereits negative Werte. Vor einer weiteren Fallunterscheidung analog zu a) bis d) muß man erst das Wurzelcluster suchen: `while (M(k) < 0) k=-M(k) ;`
4. Hat man das komplette Gitter abgearbeitet bestimmt sich N_s nach:

```
for (k=2; k<=kmax; k++)
  NS(M(k))=NS(M(k))+1;
```

4.4 Die fraktale Dimension

4.4.1 Definition der fraktalen Dimension

Die fraktale Dimension soll am Beispiel des *Sierpinski Siebs* eingeführt werden. Das Sierpinski Sieb ist ein Fraktal, das durch folgenden iterativen Prozeß entsteht:

0.) Man wählt ein Dreieck, welches die Seitenlänge $L = 1$ und die Masse $M = 1$ haben soll.

1.) Aus dreien dieser Dreiecke setzt man nun ein neues Dreieck zusammen, das in der Mitte ein Loch besitzt (siehe Abbildung). Dieses Dreieck hat dann die Seitenlänge $L = 2$ und die Masse $M = 3$.

n.) Man nimmt drei “Dreiecke” des vorigen Schritts und setzt ein neues “Dreieck” zusammen. Die Länge beträgt nun $L = 2^n$ und die Masse $M = 3^n$.

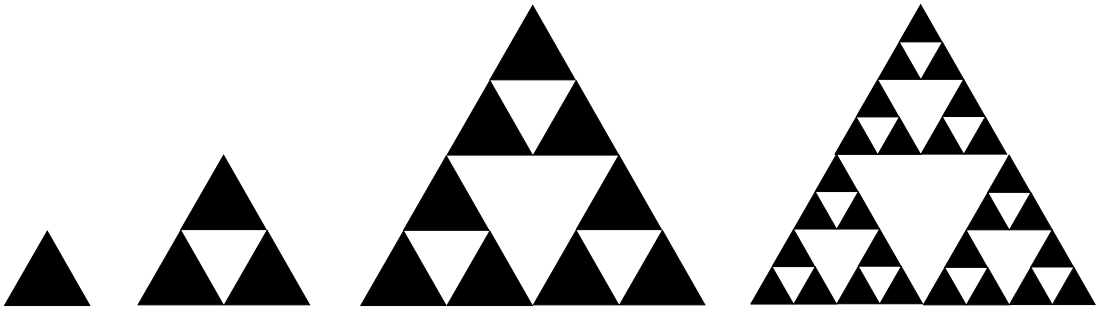


Abb. 4.5 *Konstruktion des Sierpinski Siebs*

Die fraktale Dimension wird jetzt durch die Gleichung

$$M(L) \propto L^{d_f}$$

bzw.

$$\rho = \frac{M(L)}{L^2} \propto L^{d_f - 2}$$

definiert. Das Sierpinski Sieb hat mit dieser Definition die fraktale Dimension

$$d_f = \frac{\log M(L)}{\log L} = \frac{\log 3^n}{\log 2^n} = \frac{\log 3}{\log 2} \approx 1.58$$

Betrachtet man nun Ausschnitte des Sierpinski Siebs verschiedener Größe, stellt man fest, daß das Sieb auf jeder dieser Skalen gleich aussieht. Man bezeichnet das als *Skaleninvarianz* oder *Selbstähnlichkeit*. (Das gilt natürlich nur, solange man einen Ausschnitt betrachtet, der wesentlich kleiner ist, als die Größe des kompletten Siebs und größer als ein Elementardreieck.) Für die oben betrachtete Masse bedeutet das, daß man ihre Abhängigkeit von der Systemgröße in Form eines Skalengesetzes schreiben kann:

$$M(\lambda L) = (\lambda L)^{d_f} \quad \Leftrightarrow \quad M(\lambda L) = \lambda^{d_f} M(L)$$

Eine mathematische Definition der fraktalen Dimension ist die Hausdorff-Definition. Dazu stellt man sich vor, man überdeckt das ganze Fraktal mit Scheibchen vom Radius ε so, daß die Anzahl N_ε der dazu benötigten Scheibchen minimal wird. Überdecken bedeutet dabei, daß das Fraktal an keiner Stelle mehr sichtbar sein darf. Diesen Vorgang wiederholt man für eine Reihe von Scheibchen mit $\varepsilon \to 0$. Die fraktale Dimension ist dann definiert als:

$$d_f = \lim_{\varepsilon \to 0} \frac{\ln N_\varepsilon}{\ln(L/\varepsilon)}$$

4.4.2 Anwendung auf Perkolation

Ein Algorithmus zur Berechnung der fraktalen Dimension eines Clusters nutzt die Skaleninvarianz aus ("box-counting"): Das Cluster soll auf einem Quadratgitter $N(i,j)$ sitzen mit $i,j = 1,2\ldots L$. Man rastert nun das Gitter immer gröber. Ein Punkt im groben Gitter wird gesetzt, wenn innerhalb der zugehörigen Masche mindestens ein Punkt des feinen Gitters gesetzt ist.

```
for(gridsize=1; gridsize<(L/2); gridsize++)
  {
  for(i=0; i<L; i++)
    {
    k1=(i/gridsize);
    for(j=0; j<L; j++)
      {
      k2=(j/gridsize);
      if (N[i][j]==1) M[k1][k2]=1;
      }
    }
  for(i=0; i<=L/gridsize; i++)
    {
    for(j=0; j<=L/gridsize; j++)
      {
      if (M[i][j]==1) mass[gridsize]+=1;
      }
    }
  }
```

Die logarithmische Auftragung des Quotienten aus `mass[gridsize]` und der Gesamtmasse des feinen Gitters über `gridsize` zeigt eine Gerade mit der fraktalen Dimension als Steigung.

Mit der fraktalen Dimension kann man auch eine Beziehung zwischen den oben eingeführten Exponenten ν und β herleiten. Für die Wahrscheinlichkeit, daß ein Gitterpunkt innerhalb eines Radius r mit $r < \xi$ zum unendlichen Cluster gehört, gilt:

$$P_\infty \propto \frac{r^{d_f}}{r^d}$$

d ist die euklidische Dimension des Gitters. Schreibt man nun $r = a\xi$ mit $a < 1$ erhält man:

$$P_\infty \propto \xi^{d_f - d}$$

Mit den oben eingeführten Potenzgesetzen für ξ und P_∞ erhält man:

$$\begin{aligned} (p-p_c)^\beta &\propto (p-p_c)^{-\nu(d_f-d)} \\ d_f &= d - \frac{\beta}{\nu} \end{aligned}$$

Eine entsprechende Implementation auf dem Computer trägt die Bezeichnung *sandbox-Methode*. Dazu wählt man einen beliebigen Punkt des Clusters als Ursprung. Man bestimmt nun die Anzahl M der Clusterplätze im Intervall $[r-\frac{1}{2}, r+\frac{1}{2}]$. Für die daraus

berechnete Dichte gilt dann:

$$\rho = \frac{M([r-\frac{1}{2}, r+\frac{1}{2}])}{r^d} \propto r^{d_f - d} = r^{-\frac{\beta}{\nu}}$$

Hat man also $M(r)$ bestimmt liefert die logaritmische Auftragung über r den gesuchten Exponenten.

Eine weitere Anwendung berechnet die Clustergrößenverteilung n_s aus der Skaleninvarianz: Aufgrund der Skaleninvarianz gilt die Skalentransformation:

$$\xi' = \lambda\xi \qquad s' = \lambda^{d_f} s$$

Weiter nimmt man an, daß auch auf einer anderen Skala die Anzahl der beobachteten Cluster gleich ist.

$$\begin{aligned} \int_s^\infty N_s ds &= \int_{s'}^\infty N_{s'} ds' \\ \Leftrightarrow \quad s N_s(p) &= s' N_{s'}(p') \end{aligned}$$

Mit $(p' - p_c) \propto \xi'^{-\frac{1}{\nu}} = \lambda^{-\frac{1}{\nu}} \xi^{-\frac{1}{\nu}} \propto \lambda^{-\frac{1}{\nu}} (p - p_c)$ erhält man:

$$\begin{aligned} s n_s(p - p_c) N &= \lambda^{d_f} s n_{\lambda^{d_f} s}(p' - p_c) \lambda^d N \\ \Leftrightarrow \quad n_s(p - p_c) &= \lambda^{d_f + d} n_{\lambda^{d_f} s}(\lambda^{-\frac{1}{\nu}}(p - p_c)) \end{aligned}$$

Dabei wurde noch die Beziehung $N' = \lambda^d N$ ausgenutzt. Wählt man nun $\lambda = s^{-\frac{1}{d_f}}$ (was immer möglich ist, da λ ein beliebiger Faktor ist), kann man schreiben:

$$n_s(p - p_c) = s^{-\frac{d_f + d}{d_f}} n_1(s^{\frac{1}{\nu d_f}}(p - p_c))$$

n_1 ist eine Funktion, die nicht mehr von s abhängt. Das so gewonnene Skalengesetz schreibt man einfacher als

$$n_s(p) = s^{-\tau} F(s^\sigma (p - p_c))$$

mit $\tau = \frac{d_f + d}{d_f}$ und $\sigma = \frac{1}{\nu d_f}$. Man kann dieses Gesetz überprüfen, indem man für verschiedene Skalenbereiche $n_s s^\tau$ über $s^\tau (p - p_c)$ aufträgt. Es kommt dann zu einem Datenkollaps, d.h. alle Punkte liegen auf einer Kurve.

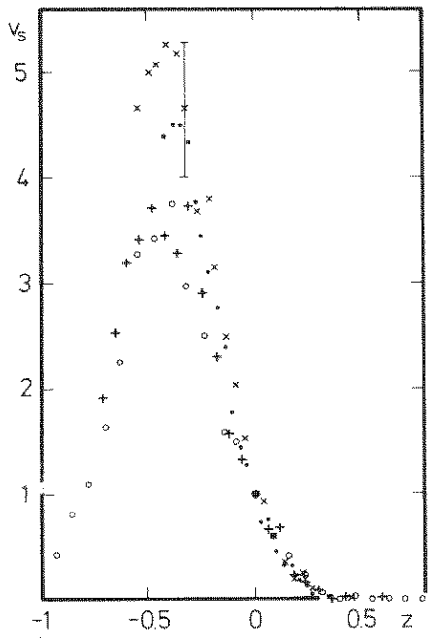


Abb. 4.6 *Test des Skalengesetzes. Verschiedene Symbole gehören zu verschiedenen Skalen.*

4.5 Finite-Size-Effekte

Betrachtet man ein Gitter endlicher Größe, was man bei einer Implementierung auf dem Computer zwangsläufig macht, dann kommt es im Bereich um p_c, für den die charakteristische Länge ξ eines Clusters dieselbe Größenordnung hat wie die Gitterabmessung L, zu finite-size-Effekten. Das bedeutet, daß die gemessenen Größen von der Gittergröße abhängig werden. Die Resultate werden falsch, wenn $\xi > L$. Möchte man z.B. die kritische Besetzungswahrscheinlichkeit p_c bestimmen und betrachtet dazu die Wahrscheinlichkeit für das Auftreten von Perkolation für verschiedene p, so würde man im Falle des unendlichen Gitters eine Sprungfunktion bei p_c messen. Bei einem endlichen Gitter verschmiert diese Funktion jedoch. Als kritische Besetzungswahrscheinlichkeit könnte man jetzt die Abszisse des Wendepunktes definieren. Man beobachtet dann eine Abhängigkeit der gemessenen kritischen Besetzungswahrscheinlichkeit $p_c(L)$ von p_c der Form $|p_c(L) - p_c| \propto L^{-\frac{1}{\nu}}$. D.h. man hat wieder ein Skalengesetz, mit dessen Hilfe man durch Messungen bei verschiedenen Gittergrößen L von endlichen Gittern auf den unendlichen Fall extrapolieren kann.

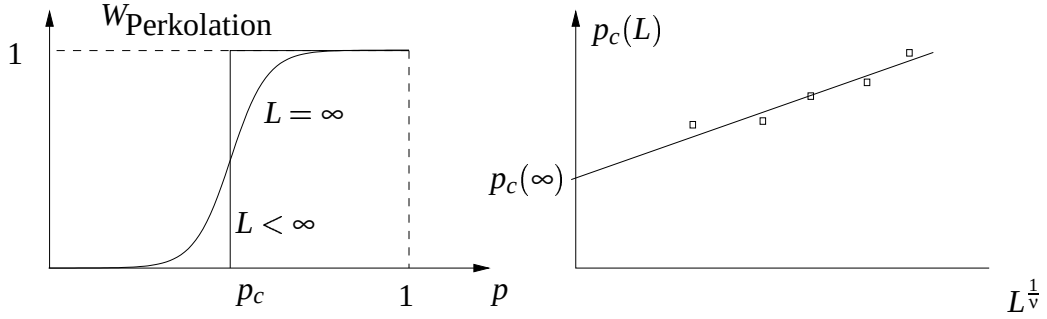


Abb. 4.7 *Bestimmung der kritischen Besetzungswahrscheinlichkeit*

4.6 Beispiele für Anwendungen der Perkolationen

1. Böden (allgemein: poröse Medien): Die besetzten Gitterplätze stellen hier Poren, d.h. Hohlräume im Boden, dar. Entsprechend bezeichnet man p hier als Porosität. p_c wäre dann die minimale Porosität, bei welcher der Boden wasserdurchlässig wird.
2. Gelbildung: Die gesetzten Gitterpunkte symbolisieren hier Monomere. p ist der Polymerisationsgrad. Dieser ist eine monotone Funktion der Zeit. Untersucht wird z.B. die kritische Zeit t_c, bei der sich zum ersten Mal ein unendlicher Cluster, sprich ein Makromolekül, gebildet hat, welches das Gel bildet und ein endliches Schermodul hat.
3. Binäre Mischungen: Hier simuliert man z.B. die Mischung von Leitern und Nichtleitern oder von Supraleitern und Normalleitern. p_c liefert in diesem Fall das kritische Mischungsverhältnis, bei welchem das Material leitend, bzw. supraleitend wird.

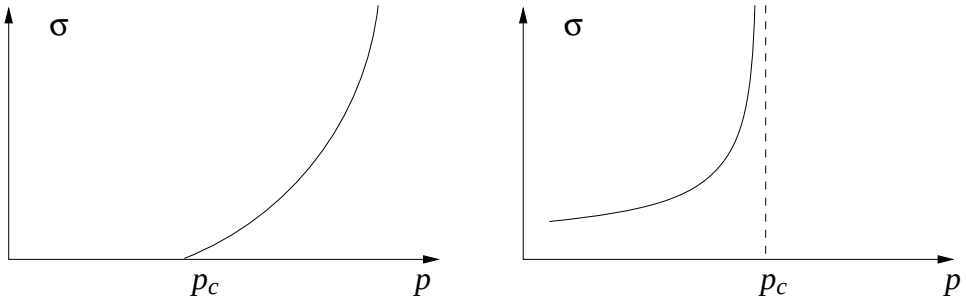


Abb. 4.8 *links: elektrische Leitfähigkeit in Abhängigkeit von p in einer Mischung aus Normal- und Nichtleiter rechts: Mischung aus Supra- und Normalleiter*

Kapitel 5

Zellularautomaten

5.1 Einführung

Zellularautomaten definiert man wie folgt: Gegeben sei ein Gitter mit N Gitterplätzen (Zellen). Jede Zelle kann r verschiedene Zustände $\sigma_i = 0,...,r-1$ annehmen. Der Zustand einer Zelle i zur Zeit $t+1$ hängt nach einer festen Regel von den Zuständen von k Zellen zur Zeit t ab. Die Regel wird bestimmt durch

$$\sigma_i(t+1) = f_i(\sigma_j(t), j = 1,\ldots,k)$$

oder allgemeiner

$$\sigma_i(t+1) = f_i(\sigma_j(t),\ldots,\sigma_j(t-\tau),z)$$

In der letzten Formulierung soll z eine Zufallszahl sein.

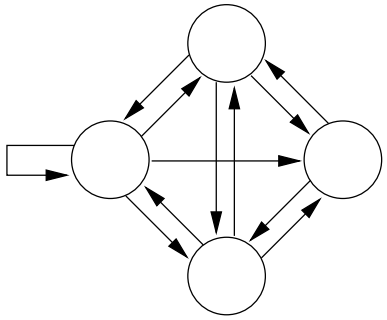


Abb. 5.1 *ZA aus vier Zellen mit je drei Eingängen (k=3)*

Damit hat ein ZA folgende Eigenschaften:

- diskreter Raum, diskrete Zeit und diskrete Anzahl der Zustände
- Es existiert eine Regel, um von t nach $t+1$ zu kommen, die von k Eingängen abhängt.
- Die Regel wird parallel auf alle Zellen angewendet.

5.2 Notation und Klassifikation

Es soll nun ein eindimensionaler ZA betrachtet werden, genauer eine Kette von N Zellen mit periodischen Randbedingungen. Jede Zelle soll zwei mögliche Zustände $(0,1)$ und drei Eingänge besitzen. Die Regel läßt sich damit schreiben als:

$$\sigma_i(t+1) = f(\sigma_{i-1}(t), \sigma_i(t), \sigma_{i+1}(t))$$

Zunächst soll eine Regel angewandt werden werden, die jeder Zelle zur Zeit $t+1$ dann eine 1 zuordnet, wenn genau einer ihrer Eingänge zur Zeit t den Zustand 1 hat. Eingänge sind die Zelle selber und ihre beiden nächsten Nachbarn. Mit dieser Regel und periodischen Randbedingungen erhält man beispielsweise folgende zeitliche Entwicklung:

$t = 0$ 10010100011
$t = 1$ 01110110100
$t = 2$ 10000000110

Betrachtet man alle möglichen Eingänge und die zugehörigen Ausgänge erhält man folgende Zuordnung:

Eingänge	111	110	101	100	011	010	001	000
$f(n)$	0	0	0	1	0	1	1	0

Diese Auftragung liefert die Zahl 22 in Binärdarstellung. Die oben formulierte Regel trägt daher den Namen “Regel 22”. Man sieht auch, daß sich in diesem Fall eines ZA mit drei Eingängen und je zwei Zuständen genau $256 = 2^{2^3}$ Regeln formulieren lassen, die man alle eindeutig in der Form

$$c = \sum_{n=0}^{7} 2^n f(n)$$

benennen (nummerieren) kann. n zählt die möglichen verschiedenen Eingangskonfigurationen durch.

Im allgemeinen Fall eines ZA (in einer Dimension) mit r Zuständen und k Eingängen hat man r^k mögliche Eingangskonfigurationen und kann r^{r^k} verschiedene Regeln formulieren. Die Benennung der Regel erfolgt nach der von Wolfram (1981) eingeführten Nomenklatur:

$$c = \sum_{n=0}^{r^{k-1}} r^n f(n)$$

c ist immer eine ganze Zahl, welche die Regel eindeutig definiert und gerade der in der Basis r dargestellten Wahrheitstafel entspricht.

Als Beispiele für die Namensgebung seien noch folgende Regeln für obigen Zellularautomaten notiert:

Eingänge	111	110	101	100	011	010	001	000
Regel 4	0	0	0	0	0	1	0	0
Regel 8	0	0	0	0	1	0	0	0
Regel 90	0	1	0	1	1	0	1	0

Tabelle 5.1: Regeln für Zellularautomaten

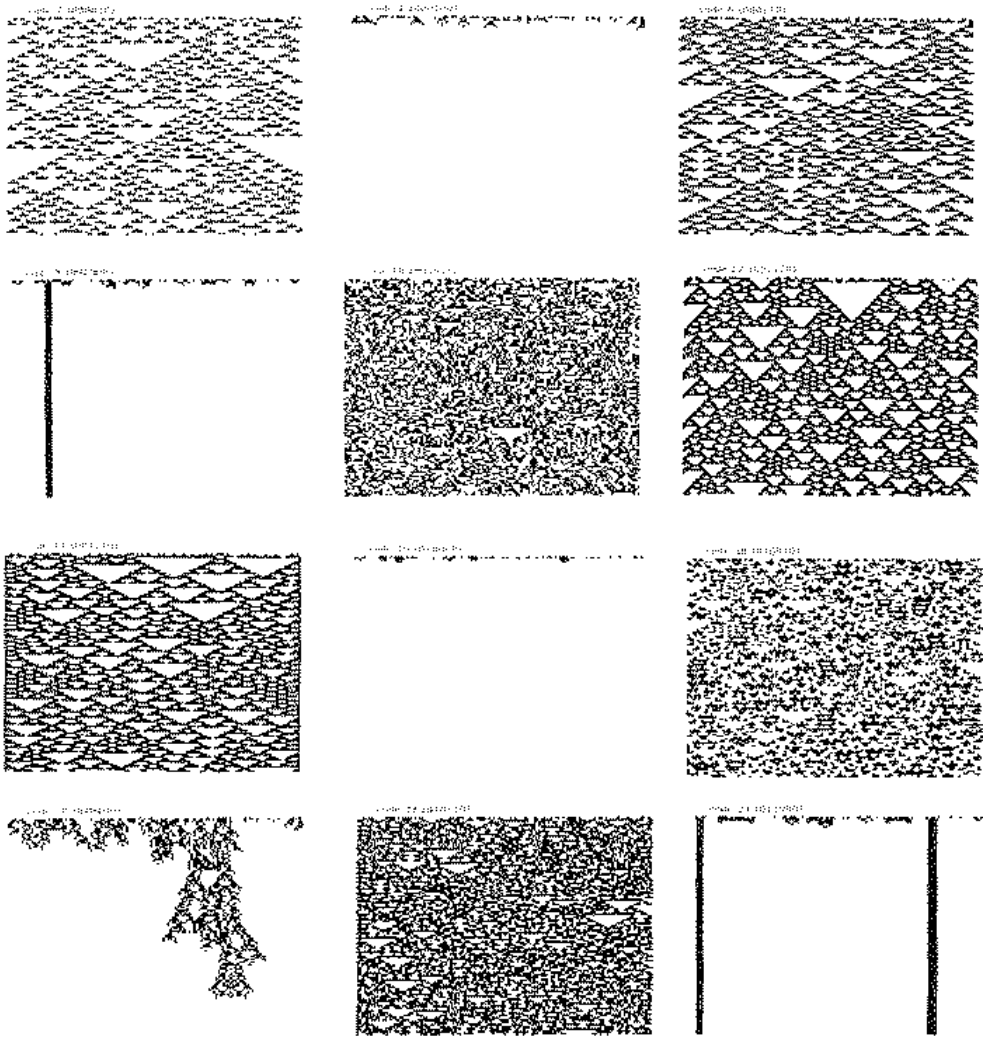


Abb. 5.2 *Zeitliche Entwicklung verschiedener Zellularautomaten mit jeweils fünf Eingängen und zwei Zuständen pro Zelle.*

Beobachtet man die zeitliche Entwicklung von ZA mit verschiedenen Regeln, kann man vier Typen bzw. Klassen unterscheiden:

1. Nach einer gewissen Zeit sind alle Einsen verschwunden (z.B. Regel 8)
2. Für lange Zeiten erhält man eine stationäre Struktur, d.h. die Zustände der einzelnen Zellen ändern sich nicht mehr oder haben kleine Perioden.

3. Der Automat verhält sich chaotisch. Das “Muster” ähnelt dem Sierpinski Sieb. (Die Regel 90 liefert mit einer einzigen Eins als Ausgangszustand genau das Sierpinski Sieb.)

4. Es bilden sich großräumige Strukturen.

Die Einteilung der ZA in vier verschiedene Klassen stammt von Wolfram. Inzwischen weiß man jedoch, daß die Automaten der 4.Klasse für sehr lange Zeiten das gleiche Verhalten zeigen wie die Automaten der 2.Klasse, jedoch wesentlich längere Transienten haben (siehe unten).

Chaotisches Verhalten von ZA liegt normalerweise dann vor, wenn zwei ursprünglich im Phasenraum dicht beieinanderliegende Konfigurationen sich exponentiell mit der Zeit voneinander entfernen. Um chaotisches Verhalten zu messen, definiert man eine Metrik im Phasenraum. Die zeitliche Entwickung der Entfernung zwischen zwei Konfiguration σ_i^A und σ_i^B wird durch den *Hammingabstand*

$$d(t) = \frac{1}{N} \sum_i \sigma_i^A \oplus \sigma_i^B$$

gemessen ($\oplus$ bezeichnet wieder das exklusive ODER). Für kleine Zeiten soll d gegen 0 gehen. Geht für $t \to \infty \quad d(\infty) \to 0$ bezeichnet man das System als eingefroren. Wird für $t \to \infty \quad d(\infty)$ endlich, bezeichnet man das System als chaotisch. In der Praxis nimmt man zwei identische Systeme und ändert in einem System lediglich eine Zelle. Nun betrachtet man die zeitliche Entwicklung dieser Störung und bestimmt jeweils $d(t)$. Ein Beispiel für die zeitliche Ausbreitung eines solchen Unterschiedes zeigt die Abbildung. Als Maß dafür, wie stark chaotisch ein System ist, dient der *Ljapunov-Exponent*. Er kann im vorliegenden Fall definiert werden als

$$\lambda = \frac{1}{\text{Steigung der Einhüllenden in der Abbildung}}$$

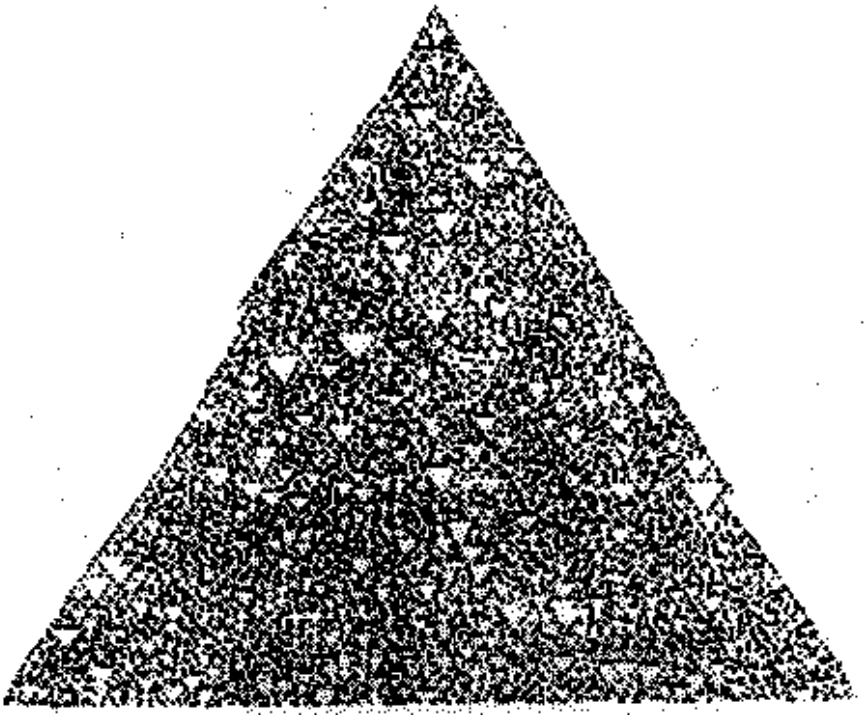


Abb. 5.3 *Ausbreitung einer Störung*

Betrachtet man das Verhalten von ZA im Phasenraum, beobachtet man Systeme, die auf einen Fixpunkt zulaufen und Systeme, die sich auf ringförmig geschlossenen Zyklen bewegen. Die auf einen solchen Grenzzyklus bzw. einen Fixpunkt zuführende Trajektorie nennt man Transient. Die Transienten werden nur einmal durchlaufen. Der Grenzzyklus wird periodisch immer wieder durchlaufen. Die Trajektorien der ZA der Klasse eins und die der Klasse zwei, die nicht periodisch sind, laufen alle auf einen Fixpunkt zu. Die Klasse eins besitzt dabei nur die Null als einzigen Fixpunkt. Das (anfänglich) andere Verhalten der ZA der vierten Klasse rührt davon her, daß diese Automaten sehr lange Transienten besitzen. Die ZA der dritten Klasse bewegen sich im Phasenraum auf sehr langen Zyklen (vergleiche mit den Generatoren für Zufallszahlen). Die periodischen Automaten der zweiten Klasse haben kurze Zyklen.

Abb. 5.4 *Mögliches Verhalten im Phasenraum*

Schließlich soll noch die zeitliche Entwicklung der “Dichte der Einsen” M betrach-

tet werden. Im Allgemeinen gilt, daß sich $M(t)$ für große Zeiten asymptotisch einem Grenzwert nähert:

$$M(t) = M(\infty)(1 - \alpha e^{-\frac{t}{\tau}})$$

τ ist wieder eine Relaxationszeit. Trägt man $M(t)$ für einen einzelnen ZA über t auf, beobachtet man starke Fluktuationen. ZA sind hier vergleichbar mit den Generatoren für Zufallszahlen.

5.3 Implementation auf dem Computer

Jede binäre Regel läßt sich auch als logische Funktion darstellen. Die obigen Regeln haben z.B. folgende Darstellung:

$$\begin{aligned} f_8(\sigma_{i-1},\sigma_i,\sigma_{i+1}) &= \bar{\sigma}_{i-1} \wedge \sigma_i \wedge \sigma_{i+1} \\ f_{22}(\sigma_{i-1},\sigma_i,\sigma_{i+1}) &= (\bar{\sigma}_{i-1} \wedge \sigma_i \wedge \bar{\sigma}_{i+1}) \vee (\sigma_{i-1} \wedge \bar{\sigma}_i \wedge \bar{\sigma}_{i+1}) \vee (\bar{\sigma}_{i-1} \wedge \bar{\sigma}_i \wedge \sigma_{i+1}) \\ f_{90}(\sigma_{i-1},\sigma_i,\sigma_{i+1}) &= (\bar{\sigma}_{i-1} \wedge \sigma_{i+1}) \vee (\sigma_{i-1} \wedge \bar{\sigma}_{i+1}) = \sigma_{i-1} \oplus \sigma_{i+1} \end{aligned}$$

Es gibt nun zwei effektive Möglichkeiten diese Regeln auf dem Computer zu implementieren:

1. *lookup-Tafeln*: In einem Feld legt man $f(n)$ ab. n nummeriert wieder die Eingänge durch. In unserem Beispiel eines ZA mit drei Eingängen und zwei Zuständen pro Zelle berechnet sich n nach

   $$n = \mathrm{shl}(\mathrm{shl}(\sigma_{i-1}) \wedge \sigma_i) \wedge \sigma_{i+1}$$

   shl bezeichnet die Operation *shift left*. Dabei werden alle Bits eines Datenwortes um eine Position nach links verschoben. In das letzte Bit rückt eine Null nach, das erste Bit fällt aus dem Datenwort heraus. Mathematisch entspricht shl einer Multiplikation mit 2. Hat man n so berechnet erhält man den Zustand der Zelle zur Zeit $t+1$ durch einen einfachen Speicherzugriff:

   $$\sigma_i(t+1) = f(n)$$

2. *multi-spin-coding*: Diese Methode kann nur im eindimensionalen Fall binärer Zustände angewandt werden. Sie hat den Vorteil, daß mehrere Zellen auf einmal behandelt werden. Erreicht wird das dadurch, daß jedes Bit eines Datenwortes D den Zustand einer Zelle repräsentiert. Nun benutzt man die Darstellung der Regel f als logische Funktion. Möchte man z.B. die Regel 90 implementieren, erreicht man das für alle in D dargestellten Zellen einfach durch die Operation

   $$D(t+1) = \mathrm{shl}(D(t)) \oplus \mathrm{shr}(D(t))$$

   (shr bezeichnet analog zu shl ein Verschieben nach rechts.) Durch die “bit-by-bit” Eigenschaft von shr und der logischen Operationen kann man dann auf einer 64-bit

Maschine 64 Gitterplätze gleichzeitig bestimmen. Die Cray braucht typischerweise 2nsec für einen Zyklus und 1-2 Zyklen für logische Operationen und shifts. Das heißt man berechnet 64 Zellen in 20-30 nsec, also einen Zellenupdate in $3 \cdot 10^{-10}$ Sekunden. Das Model läuft also mit ca 3GHz, und man kann ein System der Größe 1000^3 in einer Sekunde dreimal iterieren. Durch die multi-spin-Kodierung spart man auch Speicherplatz, so daß das 1000^3 System in ca. 15 MWorte paßt.

5.4 Beispiele für Zellularautomaten

5.4.1 Der Q2R (Vichniac 1984)

Das Kürzel Q2R steht für einen auf einem **Q**uadratgitter definierten, **r**eversiblen ZA mit **2** Zuständen $\sigma = \{0,1\}$. Als Regel für den Q2R formuliert man:

$$\sigma_{ij}(t+1) = f(x_{ij}) \oplus \sigma_{ij}(t)$$

mit

$$\begin{aligned} x_{ij} &= \sum_{nn} \sigma \\ &\equiv \sigma_{i-1j} + \sigma_{i+1j} + \sigma_{ij-1} + \sigma_{ij+1} \end{aligned}$$

und

$$f(x) = \begin{cases} 1 & \text{falls} \quad x = 2 \\ 0 & \text{falls} \quad x \neq 2 \end{cases}$$

$\sum_{nn}$ bezeichnet die Summe über die nächsten Nachbarn. Mit Hilfe von logischen Operationen formuliert, lautet die Regel:

$$\sigma(t+1) = \sigma(t) \oplus (((\sigma_N \oplus \sigma_O) \wedge (\sigma_S \oplus \sigma_W)) \vee ((\sigma_N \oplus \sigma_S) \wedge (\sigma_O \oplus \sigma_W)))$$

(N kennzeichnet den “nördlichen” Nachbarn, O den “östlichen”, ...).

Man teilt das Gitter in zwei Untergitter σ und $\hat{\sigma}$. so daß sich die Gitterpunkte immer abwechseln. Dann gehören die nächsten Nachbarn eines Gitterpunktes immer zum jeweils anderen Untergitter.

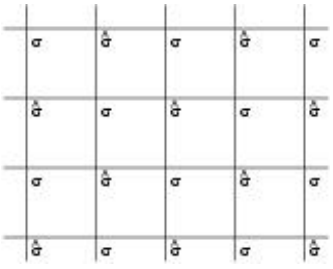


Abb. 5.5 *Aufteilung des Quadratgitters in zwei Untergitter*

Bis auf den aufzufrischenden Gitterpunkt gehen in die Regel dann nur noch Gitterpunkte des jeweils anderen Untergitters ein. Die Berechnung des Gitterzustandes zur Zeit $t+1$ erfolgt dann nicht mehr parallel, sondern es werden erst die Zellen des einen Untergitters aufgefrischt und anschließend die Zellen des zweiten Untergitters.

Die Regel soll nun so zusammengefaßt werden, daß das Auffrischen wieder parallel erfolgen kann. Dazu faßt man immer zwei benachbarte Zellen zusammen. Man erhält so ein neues Quadratgitter mit der halben Zellenzahl und vier Zuständen pro Zelle.

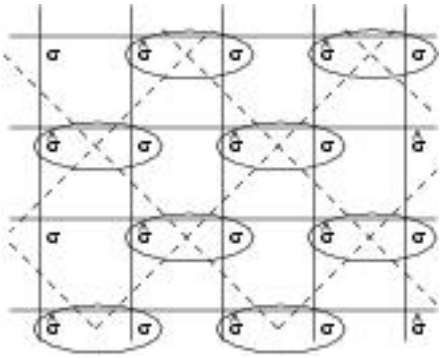


Abb. 5.6 *Zusammenfassung zweier benachbarter Gitterplätze zu einem neuen Gitterplatz mit vier möglichen Zuständen.*

Die beiden Operationen

$$A = \begin{cases} \sigma_i & = f(\hat{\sigma}_j)_{j=nn(i)} \oplus \sigma_i \\ \hat{\sigma}_i & = \hat{\sigma}_i \end{cases} \quad \text{mit} \quad AA = 1$$

und

$$B = \begin{cases} \sigma_i & = \sigma_i \\ \hat{\sigma}_i & = f(\sigma_j)_{j=nn(i)} \oplus \hat{\sigma}_i \end{cases} \quad \text{mit} \quad BB = 1,$$

die zuvor nacheinander ausgeführt wurden, faßt man zusammen zu einer Operation, die jetzt gleichzeitig ausgeführt werden kann:

$$R = AB = \begin{cases} \sigma_i & = f(\hat{\sigma}_j)_{j=nn(i)} \oplus \sigma_i \\ \hat{\sigma}_i & = f(f(\hat{\sigma}_k)_{k=nn(j)} \oplus \sigma_j)_{j=nn(i)} \oplus \hat{\sigma}_i \end{cases}$$

Die Operationen A und B vertauschen nicht. $\bar{R} = BA$ ist die zu R reversible Regel. Ein Automat f heißt reversibel, falls es einen zweiten Automaten g gibt, mit $g(\sigma(t)) = f(\sigma(-t))$. Ein reversibler Automat darf keine Transienten haben. Sein Zustandsdiagramm im Phasenraum zeigt nur geschlossene Trajektorien. Daß der Automat mit der Regel R reversibel ist, sieht man, wenn man z.B. $\bar{R}\bar{R}RR = ABABBABA$ berechnet. Man erhält den Identitätsoperator als Ergebnis. Man erkennt auch, daß sich der Automat von dem Augenblick an, an dem man das Auffrischen eines Untergitters (hier A) wegläßt, rückwärts bewegt.

Die Größe $E = \sum_{nn} \sigma_i \oplus \sigma_j$ zählt die Anzahl der Verbindungen, für die die beiden Endpunkte in verschiedenen Zuständen sind. E ist eine Erhaltungsgröße. Im Ising-Modell entspricht sie der Energie des Systems. Im Phasenraum gehört zu jedem Energiewert ein anderer Zyklus. Die Anzahl der Zyklen für ein festes E verhält sich proportional zu $2^{0.27N}$ (N ist die Anzahl der Gitterplätze). Die Länge der einzelnen Zyklen ist proportional zu $2^{0.73N}$. Der Q2R eignet sich besonders für mikrokanonische Simulationen (Simulationen bei konstanter Energie). Man hat jedoch das Problem, daß eine einzelne Simulation nicht ergodisch ist, d.h. es wird nicht der ganze Phasenraum erreicht, sondern nur die Konfigurationen eines Zyklus.

Die Größe $M = \sum_{i=1}^{N} \sigma_i$ entspricht der Magnetisierung im Ising-Modell.

Bei der Implementierung auf dem Computer wendet man die Methode des multi-spin-codings an. Hat man z.B. einen Rechner mit einer Wortlänge von 64 Bit, wählt man als Seitenlänge des Quadratgitters bevorzugt ein Vielfaches von 64 (doch mindestens 128). Es soll angenommen werden, das Gitter hätte 128×128 Zellen. Dann wird die erste Zelle der ersten Zeile repräsentiert durch das erste Bit des ersten Datenwortes ($N_{Bit}(Wort, Zeile) = N_1(1,1)$), die zweite Zelle durch das erste Bit des zweiten Datenwortes der ersten Zeile ($N_1(2,1)$), die dritte Zelle durch das zweite Bit des ersten Datenwortes ($N_2(1,1)$), ... Die zweite Zeile startet mit dem ersten Bit des *zweiten* Datenwortes der zweiten Zeile. D.h. man hat wieder dieselbe Unterteilung in zwei Untergitter $N_i(1,j)$ und $N_i(2,j)$ wie oben. Für die nächsten Nachbarn der im Datenwort $N(1,1)$ repräsentierten Zellen gilt dann (bei periodischen Randbedingungen):

$$\begin{array}{ccc} & a := N(2,128) & \\ d := \mathrm{shr}(N(2,1)) & N(1,1) & b := N(2,1) \\ & c := N(2,2) & \end{array}$$

Die Regel lautet dann:

$$N(1,1) = N(1,1) \oplus (((a \oplus b) \wedge (c \oplus d)) \vee ((a \oplus c) \wedge (b \oplus d)))$$

Diese Datenorganisation läßt sich auch auf $(n \cdot 64) \times (n \cdot 64)$ Gitter übertragen. Die ersten n Zellen einer Zeile werden dann entsprechend durch jeweils das erste Bit der n Datenworte pro Zeile repräsentiert. Der Zustand der Zelle $M(i,j)$ des ursprünglichen Gitters wird dann im k-ten Bit von $N(i - l[\frac{i-1}{l}], j)$ gespeichert, mit $k = [\frac{i-1}{l}] + 1$ und $l = \frac{L}{64}$. L ist die ursprüngliche Seitenlänge des Quadratgitters.

Die periodischen Randbedingungen implementiert man mit sogenannten *Nachbarschaftstafeln*. In diesen Tafeln ist für jede Spaltenposition die Spaltenposition des linken bzw. rechten Nachbars abgespeichert. Für ein System der Größe $L \times L$ besitzen die Nachbarschaftstafeln folgendes Aussehen:

$$K_+(i) = \begin{cases} i+1 & \text{falls } i < L \\ 1 & \text{falls } i = L \end{cases}$$

(linker Nachbar)

$$K_-(i) = \begin{cases} i-1 & \text{falls } i > L \\ L & \text{falls } i = 1 \end{cases}$$

(rechter Nachbar)
Die zu den Zellen in $N(i,j)$ benachbarten Zellen findet man damit in $N(K_{\pm}(i),j)$ und $N(i,K_{\pm}(j))$. Für das erste und letzte Datenwort muß man zusätzlich noch folgende Operation durchführen:

$$\begin{array}{lll} \mathrm{shr}(N(K_{-}(i),j)) & \text{falls} & i=1 \\ \mathrm{shl}(N(K_{+}(i),j)) & \text{falls} & i=L \end{array}$$

Unter Verwendung dieser Methoden kann das Programm so weit beschleunigt werden, daß die Bearbeitung von sehr großen Gittern möglich wird (Herrmann und Zabolitzky, 1988).

5.4.2 Gittergasmodelle

Gittergasmodelle (GGM) stellen eine alternative Methode zur Berechnung der Bewegung von Flüssigkeiten dar. Das modellierte Gas bzw. die Flüssigkeit ist dabei (wie bei allen ZA) diskret in Raum und Zeit und zusätzlich auch in den Geschwindigkeiten (siehe unten). Das einfachste GGM wurde 1986 von Frisch, Hasslacher und Pomeau eingeführt. Man betrachtet ein zweidimensionales Dreiecksgitter. Jede Verbindung in dem Dreiecksgitter hat vier mögliche Zustände:

1. Ein Teilchen fliegt nach rechts.
2. Ein Teilchen fliegt nach links.
3. In beide Richtungen fliegt ein Teilchen.
4. Auf der Verbindung befindet sich gar kein Teilchen.

Die Auffrischungsregel für diesen ZA besteht aus zwei Teilen:

a) *Propagation*, d.h. Teilchen gehen von einem Platz zu einem benachbarten Platz. Damit macht man gleichzeitig die Annahme, daß alle Teilchen den gleichen, konstanten Geschwindigkeitsbetrag $v = \frac{l}{\Delta t}$ haben. l ist die Länge einer Verbindung im Dreiecksgitter und Δt ist die Dauer eines Zeitschritts. Zudem ist die Geschwindigkeit auch in ihrer Richtung diskretisiert, da der Geschwindigkeitsvektor eines Teilchens nur in eine der sechs Gitterrichtungen zeigen kann.

b) *Kollision*, d.h. Wechselwirkung der Teilchen auf jedem Gitterplatz.

Die beiden nichttrivialen Kollisionstypen sind in der Abbildung gezeigt (Ebenfalls möglich sind natürlich auch die um $\pm 60^o$ rotierten Fälle.).

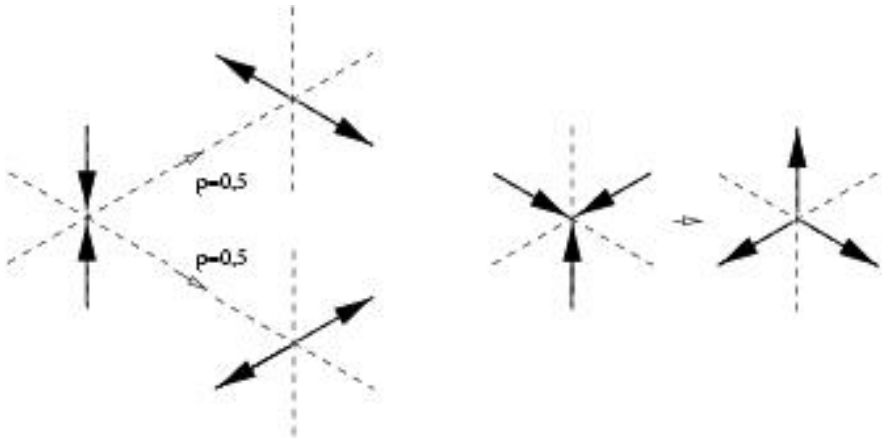


Abb. 5.7 *Mögliche Kollisionen der Teilchen eines Gittergases auf einem Dreiecksgitter*

Die erste Regel ist probabilistisch, d.h. das obere und das untere Resultat treten jeweils mit Wahrscheinlichkeit 0.5 auf. Der ZA ist in diesem Punkt also nicht deterministisch. Die Kollision zweier Teilchen, bei der jedes Teilchen in die Richtung, aus der es gekommen ist zurückfliegt, braucht nicht betrachtet zu werden, da dieser Fall identisch ist mit zwei Teilchen, die sich gegenseitig gar nicht spüren und einfach durcheinander hindurchfliegen. In allen anderen Fällen, die nicht einer der beiden obigen Kollisionskonfigurationen entsprechen, fliegen die Teilchen durch den Gitterplatz durch, ohne miteinander wechselzuwirken.

Die zweite Regel ist wichtig, um zu verhindern, daß der Impuls in jeder Gitterrichtung einzeln erhalten wird. Insgesamt gilt jedoch für das Gesamtsystem sowohl Impuls- als auch Energie- und Teilchenzahlerhaltung.

Bei Kollisionen mit der Wand benutzt man eine *no-slip* Bedingung. Makroskopisch bedeutet das, daß die Geschwindigkeit der Flüssigkeit an der Wand gleich Null ist. Die Teilchen des Gittergases werden an der Wand einfach reflektiert (Umkehrung der Komponenten senkrecht zur Wand). Sie geben dabei Impuls an die Wand ab.

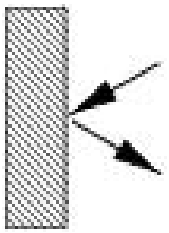


Abb. 5.8 *Kollision mit der Wand*

Programmiert wird das Gittergas als Automat mit $k = 6$ Eingängen und $r = 2^6$ Zuständen. Von den vier Züstanden einer Verbindung gehören zwei zu jedem Gitterpunkt. $\sigma_i^j = 1$ soll dabei bedeuten, daß auf der j-ten Verbindung des i-ten Gitterpunktes ein Teilchen vom Gitterpunkt wegfliegt. Für $\sigma_i^j = 0$ ist kein wegfliegendes Teilchen auf dieser Verbindung vorhanden.

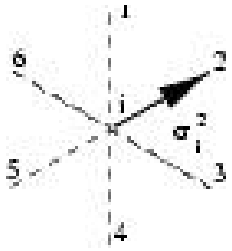


Abb. 5.9 *Benennung der Verbindungen im Gittergasmodell*

Der einzige freie Parameter dieses besonders einfachen Modells ist die Dichte der Teilchen. Die Teilchendichte auf einem Gitterpunkt definiert man als:

$$\rho_i = \frac{1}{6}\sum_{j=1}^{6}\sigma_i^j$$

Die Geschwindigkeit auf einem Gitterpunkt ist gegeben als:

$$\vec{v}_i = \frac{1}{6}\sum_{j=1}^{6}\vec{n}^j\sigma_i^j$$

$\vec{n}^j$ ist der Einheitsvektor in Richtung j.

Um makroskopische Meßgrößen zu erhalten, führt man eine Mittelung, das sogenannte *coarse graining*, durch. Dazu definiert man Kästchen, die größer sind als die Gittermaschen des Dreiecksgitters und mittelt in diesen Kästchen.

$$\vec{v} = \frac{\sum_i \vec{v}_i}{\sum_i \rho_i} \qquad \text{mit } i \in \text{Kästchen}$$

Durch Vergröberung des Gitters, in dem gemittelt wird, kann man beliebig kontinuierliche Geschwindigkeitsverteilungen modellieren. Auf diese Weise kann man Strömun-

gen wie die in der Abbildung gezeigte von Karmanstraße simulieren.

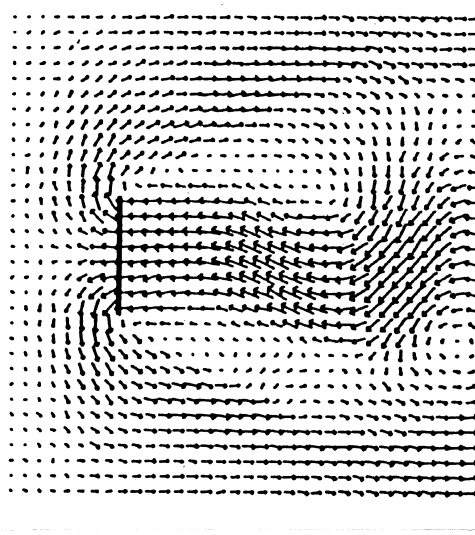


Abb. 5.10 *Simulation einer von Ka*

Die Einordnung der GGM im Vergleich zu anderen Techniken, wie Molekulardynamik (MD), finiten Elementen (FEM) und Gitter-Boltzmannmodellen (LBM) zeigt die Abbildung.

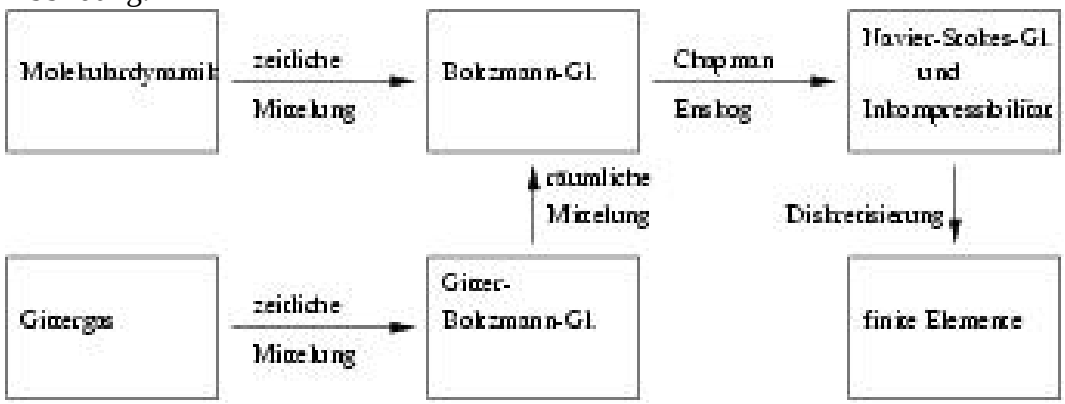


Abb. 5.11 *Einordnung der GGM im Vergleich zu anderen Techniken*

Beim Übergang vom GGM zur Gitter-Boltzmanngleichung wird eine zeitliche Mittelung durchgeführt. Dazu definiert man eine Teilchendichte in eine Gitterrichtung.

$$r_i^j = \frac{1}{T}\sum_{t=1}^{T}\sigma_i^j(t)$$

Für die $\vec{r}_i$ kann man dann eine Bewegungsgleichung aufstellen:

$$r_i^j(t+1) = r_i^j(t) + \lambda(r_{i,0}^j - r_i^j(t))$$

$r_{i,0}$ ist die Gleichgewichtsverteilung. (Für weitere Einzelheiten siehe: R. Benzi, S. Succi und M.Vergassola; Phys. Rep. 222, 145(1992)). Der Vorteil der Gittergasmethode

gegenüber der Methode der finiten Elemente besteht darin, daß mit binären Zahlen gerechnet wird, d.h. es daß keine Rechenungenauigkeiten entstehen.

Kapitel 6

Die Monte-Carlo-Methode

6.1 Einführung

Es gibt eigentlich nicht *die eine* Monte-Carlo-Methode, sondern dieser Begriff bezeichnet i.a. numerische Algorithmen, die Zufallszahlen zur Simulation oder approximativen Lösung eines komplexen Problems einsetzen. Es ist für solche stochastische Algorithmen charakteristisch, daß sie

- häufig die einzigen (realistischen) Simulationsmethoden darstellen, die in gegebener Rechenzeit ein brauchbares Resultat liefern können,
- unter Einsatz von mehr Rechenzeit systematisch verbesserbar sind,
- sie in vielen Bereichen einsetzbar sind, z.B.
  (i) sind sie für fast alle Probleme geeignet, die stochastische Elemente enthalten, z.B. bei der Berechnung von Eigenschaften ungeordneter Medien, bei Systemen der Gleichgewichtsstatistik, in denen thermische Bewegung eine Rolle spielt, beim Durchgang von (Teilchen)Strahlung durch Materie, bei Stoffumwandlungen, bei Warteschlangenproblemen, etc.;
  (ii) ebenso für eigentlich "analytische" Fragestellungen, wie der Auswertung hochdimensionaler Integrale, oder bestimmter Typen von Differentialgleichungen unter komplexen Randbedingungen (z.B. Poisson-Gleichung mit bewegten Rändern).

Der Name Monte-Carlo-Methode rührt von der Stadt Monte-Carlo und dem dortigen Spielkasino her, in dem auch mit Zufallszahlen "gearbeitet" wird.

6.2 Der $M(RT)^2$ Algorithmus

Wir betrachten ein kanonisches Ensemble, d.h. ein System mit einer konstanten Temperatur T (Ankopplung an ein Wärmebad). Die Wahrscheinlichkeit dafür, daß sich das System in einem Zustand C mit der Energie E_C befindet, beträgt dann:

$$p_{eq}(C) = \frac{1}{Z_T} \exp(-\frac{E_C}{kT})$$

Den Normierungsfaktor Z_T bezeichnet man als Zustandssumme. Er ergibt sich einfach aus der Bedingung, daß die Summe über alle Wahrscheinlichkeiten gleich 1 sein muß.

$$\sum_C p_{eq}(C) = 1 \qquad \Leftrightarrow \qquad Z_T = \sum_C \exp(-\frac{E_C}{kT})$$

Die Wahrscheinlichkeitsverteilung $p_{eq}(E)$ ist für zunehmende Systemgröße sehr stark um eine mittlere Energie $< E >_T$ konzentriert.

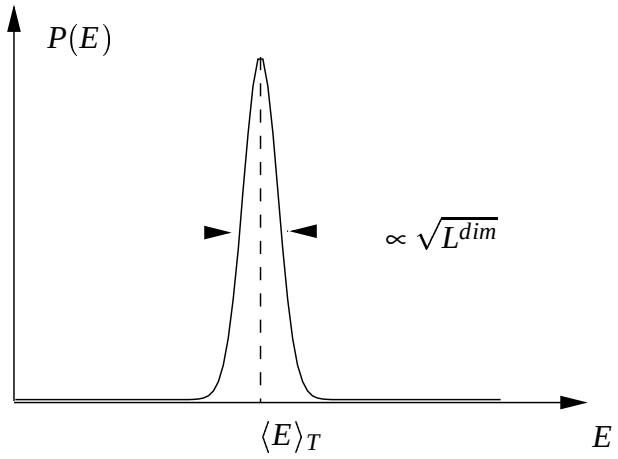


Abb. 6.1 *Wahrscheinlichkeit eines Zustandes in Abhängigkeit von seiner Energie*

Nur wenig entfernt von $< E >_T$ ist die Wahrscheinlichkeit fast Null (siehe Abbildung). Möchte man nun den Erwartungswert einer Größe $\bar{A}_T$ bestimmen mit

$$\bar{A}_T = \sum_C A(C) p_{eq}(E_C),$$

dann würde man, sofern man zufällige Konfigurationen verwendet, die über der Energie gleichverteilt sind, viel Zeit damit verschwenden, Summenglieder zu berechnen, für die $p_{eq}(E_C) \approx 0$ ist. Man hat also ein Problem, das äquivalent dazu ist, Zufallszahlen mit einer vorgegebenen Verteilung erzeugen zu müssen. Eine Lösung stammt von Metropolis, Rosenbluth, Rosenbluth, Teller und Teller (1953) und wird als $M(RT)^2$ Algorithmus bezeichnet. Es wird eine Folge $X_1 \to X_2 \to \ldots$ von Systemzuständen erzeugt, wobei jeder Zustand nur von seinem direkten Vorgänger abhängig ist. Man nennt dies einen *Markov*-Prozeß (Das entspricht einem random-walk im Phasenraum). Für die Wahrscheinlichkeit eines einzelnen Zustandes soll gelten:

$$p(\vec{X}) = p_{eq}(\vec{X})$$

Um ausgehend von einem aktuellen Zustand $\vec{X}$ in einen Folgezustand $\vec{Y}$ zu gelangen, geht man nach folgendem Rezept vor:

1. Man macht einen beliebigen Zugvorschlag in einen neuen Zustand Y. Die Gesamtheit der Zugvorschläge muß dabei so geartet sein, daß das System in endlich vielen Zügen von jedem beliebigen Zustand $\vec{X}$ in alle anderen Zustände $\vec{X}'$ gelangen kann. Diese Eigenschaft ist in unserem Zusammenhang mit der *Ergodizität* des $M(RT)^2$ Algorithmus' gleichbedeutend. Wir bezeichnen die Wahrscheinlichkeit eines Zugvorschlages nach $\vec{Y}$ aus dem Zustand $\vec{X}$ mit $T(\vec{X} \to \vec{Y})$ vom englischen *Trail*. Die T's sind normiert, d.h. mit Sicherheit bleibt das System im Zustand $\vec{X}$ oder geht in einen anderen Zustand $\vec{Y}$ über:

$$\sum_Y T(\vec{X} \to \vec{Y}) = 1$$

   Außerdem soll $T(\vec{X} \to \vec{Y}) = T(\vec{Y} \to \vec{X})$ gelten.
2. Nun verwerfe oder akzeptiere man diesen Zugvorschlag mit einer Akzeptanzwahrscheinlichkeit $A(\vec{X} \to \vec{Y})$.
   Die Wahrscheinlichkeit für den Übergang $\vec{X} \to \vec{Y}$ beträgt also insgesamt:

$$W(\vec{X} \to \vec{Y}) = T(\vec{X} \to \vec{Y})A(\vec{X} \to \vec{Y})$$

Die zeitliche Änderung der Wahrscheinlichkeit dafür, daß sich das System in einem Zustand $\vec{X}$ befindet, kann man schreiben als:

$$\frac{dp(\vec{X},t)}{dt} = \sum_{\vec{Y}} p(\vec{Y})W(\vec{Y} \to \vec{X}) - \sum_{\vec{Y}} p(\vec{X})W(\vec{X} \to \vec{Y})$$

Diese Gleichung bezeichnet man als *Mastergleichung*. Der erste Term der rechten Seite ist die Wahrscheinlichkeit dafür, daß sich das System im Zustand $\vec{Y}$ befindet und von dort nach $\vec{X}$ übergeht. Der zweite Term ist die Wahrscheinlichkeit dafür, daß sich das System im Zustand $\vec{X}$ befindet und diesen verläßt. Wir suchen die Gleichgewichtsverteilung des Systems, d.h. den stationären Zustand, in dem gilt:

$$\frac{dp(\vec{X},t)}{dt} = 0 \qquad \Leftrightarrow \qquad p(\vec{X}) = p_{eq}(\vec{X})$$

Eingesetzt in die obige Gleichung erhält man damit:

$$\sum_{\vec{Y}} p_{eq}(\vec{Y})W(\vec{Y} \to \vec{X}) = \sum_{\vec{X}} p_{eq}(\vec{X})W(\vec{X} \to \vec{Y})$$

Diese Gleichung ist bestimmt erfüllt, wenn sie für jeden Summanden einzeln erfüllt ist.

$$p_{eq}(\vec{Y})W(\vec{Y} \to \vec{X}) = p_{eq}(\vec{X})W(\vec{X} \to \vec{Y})$$

Das ist die Bedingung des *detailed balance*. Sie besagt, daß im Gleichgewicht die Anzahl der Übergänge $\vec{X} \to \vec{Y}$ gleich der von $\vec{Y} \to \vec{X}$ ist. Diese Bedingung ist hinreichend aber nicht notwendig.

Metropolis et al. schlägt nun folgende Wahl der Akzeptierwahrscheinlichkeit vor:

$$A(\vec{X} \to \vec{Y}) = \min(1, \frac{p_{eq}(\vec{Y})}{p_{eq}(\vec{X})})$$

Es soll nun gezeigt werden, daß diese Wahl die Bedingung des detailed balance erfüllt.

$$\begin{aligned} p_{eq}(\vec{Y})T(\vec{Y} \to \vec{X})A(\vec{Y} \to \vec{X}) &= p_{eq}(\vec{X})T(\vec{X} \to \vec{Y})A(\vec{X} \to \vec{Y}) \\ \Rightarrow \quad p_{eq}(\vec{Y})A(\vec{Y} \to \vec{X}) &= p_{eq}(\vec{X})A(\vec{X} \to \vec{Y}), \end{aligned}$$

wobei $T(\vec{X} \to \vec{Y}) = T(\vec{Y} \to \vec{X})$ ausgenutzt wurde. Es soll nun o.B.d.A. angenommen werden, daß $p_{eq}(\vec{Y}) > p_{eq}(\vec{X})$. Mit der obigen Wahl der Akzeptierwahrscheinlichkeit ergibt sich somit:

$$p_{eq}(\vec{Y}) \cdot \frac{p_{eq}(\vec{X})}{p_{eq}(\vec{Y})} = p_{eq}(\vec{X}) \cdot 1$$

Damit wäre die Gültigkeit des detailed balance bewiesen.

Mit $p_{eq}(\vec{X}) = \frac{1}{Z_T} \exp(-\frac{E(\vec{X})}{kT})$ kann man die Akzeptierwahrscheinlichkeit auch schreiben als:

$$A(\vec{X} \to \vec{Y}) = \min(1, \exp(-\frac{E(\vec{Y}) - E(\vec{X})}{kT}) = \min(1, \exp(-\frac{\Delta E}{kT}))$$

Ein Zugvorschlag, bei dem die Energie erniedrigt wird, wird somit immer akzeptiert. Ein Zugvorschlag, der die Energie erhöht, wird nur mit der Wahrscheinlichkeit $\exp(-\frac{\Delta E}{kT})$ akzeptiert.

Eine andere Methode, als der hier beschrieben $M(RT)^2$ Algorithmus, ist die *Glauber*-Dynamik. Hier trifft man die Wahl

$$A(\vec{X} \to \vec{Y}) = \frac{\exp(-\frac{\Delta E}{kT})}{1 + \exp(-\frac{\Delta E}{kT})}$$

Auch die Glauber-Dynamik erfüllt die Bedingung des detailed balance, was man wie folgt zeigen kann:

$$\frac{W(\vec{X} \to \vec{Y})}{W(\vec{Y} \to \vec{X})} = \frac{\exp(-\frac{\Delta E}{kT})(1 + \exp(\frac{\Delta E}{kT}))}{(1 + \exp(-\frac{\Delta E}{kT}))\exp(\frac{\Delta E}{kT})} = \exp(-\frac{\Delta E}{kT}) = \frac{p_{eq}(\vec{Y})}{p_{eq}(\vec{X})}$$

Zum Abschluß sollen noch der Zusammenhang mit den Begriffen der Mathematik hergestellt werden:

1. Die Forderung nach der Gültigkeit von $W(\vec{X} \to \vec{Y}) > 0$ für alle $\vec{X}$ und $\vec{Y}$, nennt man *Ergodizitätsbedingung*.
2. $\sum_{\vec{Y}} W(\vec{X} \to \vec{Y}) = 1$ ist die Bedingung der *Normalität*.
3. $\sum_{\vec{Y}} p_{eq}(\vec{Y})W(\vec{Y} \to \vec{X}) = p_{eq}(\vec{X})$ ist die Bedingung der *Homogenität*.

6.3 Beispiel: Das Ising-Modell

Das Ising-Modell (1925) beschreibt das Verhalten eines Systems wechselwirkender Spins. Die Modellierung erfolgt auf einem Gitter. Jeder Gitterpunkt repräsentiert einen Spin mit zwei möglichen Zuständen $s_i = \pm 1$. Die Energie des Systems sei gegeben durch

$$E = -J \sum_{i,j=nn(i)} s_i s_j$$

$\sum_{nn(i)}$ sei die Summe über die nächsten Nachbarn von s_i. Einen negativen Beitrag zur Gesamtenergie erhält man, wenn benachbarte Gitterplätze im gleichen Zustand sind. Das System hat somit zwei Grundzustände: Einen, in dem alle Gitterplätze im Zustand $+1$ sind (Spins nach oben), und einen, in dem alle Gitterplätze im Zustand -1 sind (Spins nach unten). Im Folgenden betrachten wir das Isingmodell auf dem Quadratgitter.

Beim Q2R hatte man die möglichen Zustände $\sigma_i \in \{0,1\}$ und die Energie

$$E_{Q2R} = \sum_{i,j=nn(i)} \sigma_i \oplus \sigma_j.$$

Mit $s_i = 2\sigma_i - 1$ kann man beide Modelle ineinander überführen. Setzt man die letzte Beziehung in den Ausdruck für die Energie des Ising-Modells ein und expandiert man im Ausdruck für die Energie des Q2R den XOR-Operator, erhält man für den Zusammenhang der Energien:

$$\begin{aligned} E_{Q2R} &= \sum_{nn} (\sigma_i + \sigma_j - 2\sigma_i\sigma_j). \\ E &= J \sum_{nn} (2\sigma_i + 2\sigma_j - 4\sigma_i\sigma_j - 1) \\ \Rightarrow E &= 2JE_{Q2R} - JN \end{aligned}$$

Das bedeutet, daß beide Modelle bis auf eine konstante Energieverschiebung gleich sind. Insbesondere liefern sie also die gleiche Physik.

Die Magnetisierung einer Systemkonfiguration soll definiert werden als

$$M(\vec{X}) = \sum_i s_i$$

Der Erwartungswert der Magnetisierung beträgt dann $\bar{M}_T = \sum_i M(\vec{X}) p_{eq}(\vec{X})$.

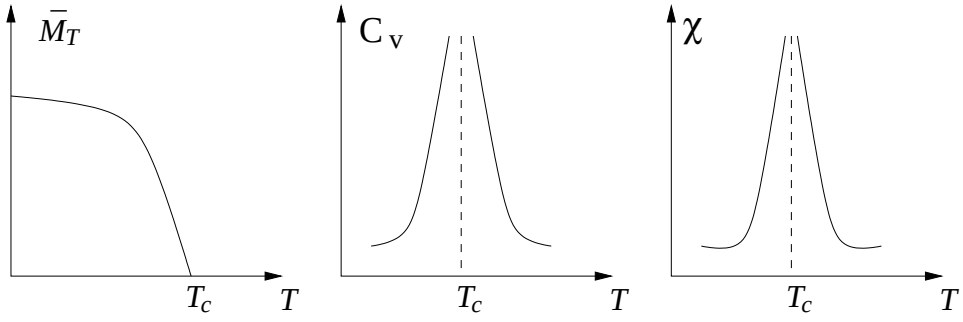


Abb. 6.2 *links: Magnetisierung Mitte: spezifische Wärme rechts: Suszeptibilität*

Die Abbildung zeigt Magnetisierung $\bar{M}_T$, spezifische Wärme $C_V = (\frac{\partial E}{\partial T})_H$ und Suszeptibilität $\chi = (\frac{\partial M}{\partial H})_T$ in Abhängigkeit von der Temperatur. Das jeweilige Verhalten in der Nähe der kritischen Temperatur wird wieder durch Potenzgesetze beschrieben:

	Potenzgesetz	2D	3D
Magnetisierung	$(T-T_c)^{\beta}$	$\beta = 1/8$	$\beta = 0.32$
spezifische Wärme	$\lvert T-T_c\rvert^{-\alpha}$	$\alpha = 0$	$\alpha = 0.11$
Suszeptibilität	$\lvert T-T_c\rvert^{-\gamma}$	$\gamma = 7/4$	$\gamma = 1.27$

Bei der Simulation des Ising-Modells mit dem Metropolis-Monte-Carlo-Verfahren entspricht ein Zug einem Spinumklappen. Zwei aufeinanderfolgende Konfigurationen in der Markov-Kette unterscheiden sich genau durch ein Spinumklappen. Der Energieunterschied zwischen zwei aufeinanderfolgenden Konfigurationen beträgt dann (unter der Annahme, daß s_i umgeklappt wurde):

$$\Delta E = J \sum_{j=nn(i)} (s_i s_j + s_i s_j) = 2Js_i \sum_{j=nn(i)} s_j =: 2Js_i h_i$$

mit $h_i = \sum_{j=nn(i)} s_j$ Man hat also folgendes Vorgehen:

1. Man wählt einen Gitterpunkt aus, dessen Spin umgeklappt werden soll. Die Auswahl kann zufällig oder regelmäßig (*type-writer*-System) erfolgen.
2. Berechnung von h_i
3. Bestimmung des Boltzmanngewichts $p_i = \exp(-2\beta s_i h_i)$ mit $\beta = \frac{J}{kT}$
4. Falls $\Delta E < 0$ ist, führt man den Zug durch. Falls $\Delta E > 0$ akzeptiert man ihn mit der Wahrscheinlichkeit p_i.

Bei der Anwendung der Glauber-Dynamik akzeptiert man im 4.Schritt den Zug mit der Wahrscheinlichkeit

$$A_i = \frac{\exp(-2\beta s_i h_i)}{1+\exp(-2\beta s_i h_i)}$$

Dabei ergibt sich die neue Konfiguration s_i' aus der alten Konfiguration s_i nach

$$s_i' = -s_i \,\mathrm{sign}(A_i - z)$$

z ist eine Zufallszahl. Mit der Definition $\tilde{p}_i := \frac{\exp(2\beta h_i)}{1+\exp(2\beta h_i)}$ schreibt man für die Wahrscheinlichkeit eines Spinumklappens:

$$p_{flip} = A_i(s_i) = \begin{cases} \tilde{p}_i & \text{für} \quad s_i = -1 \\ 1-\tilde{p}_i & \text{für} \quad s_i = +1 \end{cases}$$

Für die Erhaltung des Spins gilt dann:

$$p_{noflip} = 1 - A_i(s_i) = \begin{cases} 1-\tilde{p}_i & \text{für} \quad s_i = -1 \\ \tilde{p}_i & \text{für} \quad s_i = +1 \end{cases}$$

Die Wahrscheinlichkeit einen Spin im Zustand $+1$ zu erhalten beträgt also in beiden Fällen $\tilde{p}_i$. Analog ist die Wahrscheinlichkeit einen Spin im Zustand -1 anzutreffen $1-\tilde{p}_i$. Statt in einer Reihe von Konfigurationen immer einzelne Spins umzuklappen, kann man die Spins auch sofort mit diesen Wahrscheinlichkeiten setzen. Das ist die sogenannte "*heat-bath*"-Methode.

6.4 Implementierung auf dem Computer

6.4.1 Look-up-Tafeln

$h_i = \sum_{j=nn(i)} s_j$ kann nur fünf mögliche Werte annehmen:

$$h_i = 0, \pm 2, \pm 4$$

Für die Energiedifferenz $\Delta E = 2Js_ih_i$ ergeben sich damit folgende möglichen Werte:

$$\frac{\Delta E}{J} = 2s_ih_i = 0, \pm 4, \pm 8$$

Geht man nach obigem Schema vor und hat h_i und ΔE bereits berechnet, dann muß man beim Metropolis-Verfahren im dritten Schritt das Boltzmanngewicht nur dann bestimmen, falls $\Delta E > 0$. Im Fall $\Delta E < 0$ gilt ja $\min(1, \exp(-\Delta E/kT)) = 1$. Die verbliebenen drei Boltzmanngewichte legt man in einer Look-Up-Tafel ab:

$$P(I) = \exp(-\frac{4J}{kT}I) \quad \text{mit} \quad I = \frac{1}{2}s_ih_i \quad \in \{0,1,2\}$$

Bei der Glauber-Dynamik hat die Look-up-Tafel folgendes Aussehen:

$$P(I) = \frac{\exp(-\frac{4J}{kT}(I-3))}{1+\exp(-\frac{4J}{kT}(I-3))} \quad \text{mit} \quad I = \frac{1}{2}s_ih_i + 3 \quad \in \{0,\ldots,5\}$$

6.4.2 Multispincoding

Man packt wieder mehrere Gitterplätze in ein Datenwort. Als Beispiel soll ein dreidimensionales Ising-Modell auf einem kubischen Gitter betrachtet werden. Die Gitterzustände $s_i = \pm 1$ werden mit $\sigma_i = 0.5(s_i+1) \in \{0,1\}$ in binäre Zustände umgerechnet. Die Datenwörter sollen eine Länge von 64 Bit haben.

In drei Dimensionen hat jeder Gitterplatz sechs nächste Nachbarn. Die Energie pro Gitterplatz kann dann Werte von $0\ldots6$ annehmen. Zum Speichern der Energien muß man daher für jeden Gitterplatz drei Bits reservieren. (Auch im Gitter selbst reserviert man drei Bit pro Speicherplatz, obwohl es dort eigentlich nicht nötig ist.) Ein Datenwort representiert also immer 21 Gitterplätze. Die Datenwörter sollen wieder so organisiert sein, daß benachbarte Plätze in unterschiedlichen Datenwörtern abgelegt sind. Mit

$$E = N \,\mathrm{XOR}\, N_1 + \cdots + N \,\mathrm{XOR}\, N_6$$

berechnet man also gleichzeitig die Energien aller 21 in N abgelegten Gitterplätze. ($N_1 \ldots N_6$ enthalten jeweils die 6 nächsten Nachbarn.) Hat man die Energie berechnet durchläuft man folgende Schleife:

```
cw = 0;
for(i=1; i<=21; i++)
  {
  z=ranf();
  if(z < P(E & 7)) cw = (cw | 1); /* & = AND, | = OR */
  cw = ror(cw, 3);
  E = ror(E, 3);
  }
cw = ror(cw, 1);
N = (N ^ cw); /* ^ = XOR */
```

`z` ist eine Zufallszahl. Die Funktion `ror(E, 3)` hat als Rückgabewert E zyklisch um drei Bits nach rechts verschoben. `P(I)` ist die oben beschriebene Look-up-Tafel mit `P(I) = exp(-4JI/kT)`. `E & 7` maskiert die niederwertigsten drei Bits aus. So kann man einzeln auf die Energien der 21 Gitterplätze zugreifen. `cw` ist ein sogenanntes *changer word*. Nach dem Durchlaufen der Schleife enthält es für jeden Gitterplatz, der geändert werden soll, eine 1.

6.4.3 Kawasaki-Dynamik

Man stelle sich eine binäre Mischung zweier Konstituenten A und B vor, wie z.B. eine Legierung. Dieses System läßt sich als "Lattice-Gas" beschreiben indem man die Teilchen auf ein Gitter setzt. Seien die Energien, wenn zwei Teilchen der Sorte A nebeneinander liegen, E_{AA} und analog für die Paare BB sei die Energie E_{BB} und für die Paare AB sei sie E_{AB}. Sei ausserdem $E_{BB} = E_{AA} < E_{AB} = E_{BA}$, dann lässt sich dieses binäre System durch ein Ising-Modell beschreiben. Man identifiziert A mit dem Zustand 0 und B mit dem Zustand 1 (binäre Darstellung) und setzt $E_{AA} = E_{BB} = 0 \quad \text{und} \quad E_{AB} = 1$. Die Teilchenerhaltung impliziert allerdings, daß die Magnetisierung festgehalten werden muß.

Eine kanonische Simulation eines solchen Systems realisiert die Kawasaki-Dynamik. Charakteristisches Merkmal ist die Erhaltung der Magnetisierung. Man führe folgende Schritte durch:

1. Wähle zufällig auf dem Gitter eine Verbindung mit Zuständen AB. Nur solche Paare können geflippt werden, d.h. aus AB wird BA, da sonst die Magnetisierung nicht erhalten bliebe.

2. Man bestimme die Energie der Konfiguration der beiden Plätze AB und ihrer Umgebung von nächsten Nachbarn. Das wären auf dem Quadratgitter 6 nächste Nachbarn.

3. Akzeptieren des Paarflippens; sei ΔE der Energieunterschied, d.h. Energie nach minus Energie vor flippen.

- Metropolis-Methode, falls $\Delta E \leq 0$, dann flippen. Falls $\Delta E > 0$, flippe mit Wahrscheinlichkeit $p(I) = e^{\frac{-2I}{kT}}$, $I = \frac{\Delta E}{2} = 1,2,3$.
- Glauber Dynamik. Flippe mit Wahrscheinlichkeit

$$p(I) = \frac{e^{\frac{-2(I-3)}{kT}}}{1+e^{\frac{-2(I-3)}{kT}}} \quad , \quad I = \frac{\Delta E}{2} + 3, \quad I = 0,\ldots,6$$

6.4.4 Mikrokanonische Simulation

Mikrokanonisch heisst, daß die Energie während der Simulation erhalten bleibt. Solche Simulationen sind im Ergebnis den kanonischen äquivalent, doch in manchen Fällen effizienter. Hierzu hat Creutz (1983) eine Methode vorgeschlagen, welche mit "Dämonen" arbeitet. Jeder Dämon hat eine Energiereserve E_d, welche maximal einen Wert E_{max} erreichen kann. Man generiert aus einer Konfiguration eine neue Konfiguration, so daß die Gesamtenergie $E_t = E_s + E_d$ konstant ist. Dabei ist E_s die Energie der Konfiguration, also im Falle des Ising-Modells $E_s = -J\sum_{ij} s_i s_j$. Nun geht man folgendermassen vor:

1. Man wähle (zufällig) einen Platz.
2. Man berechne die Energiedifferenz ΔE_s zwischen der Konfiguration mit geflipptem Spin und der alten Konfiguration.
3. Falls $E_{max} \geq E_d - \Delta E_s \geq 0$, dann akzeptiere die geflippte Konfiguration.

Diese Methode hat mehrere Vorteile. Zum einen ist sie deterministisch, sodaß keine Nebeneffekte von Zufallszahlengeneratoren existieren. Die Implementierung der Methode läßt sich außerordentlich gut parallelisieren, da man alle Spins völlig parallel behandeln kann. Dazu benutzt man im allgemeinen 32 oder 64 Dämonen. Man kann auch zeigen, daß die Methode reversibel ist. Der Fall $E_{max} = 0$ entspricht Q2R und hat bekanntlich Probleme mit der Ergodizität. Eine Schwierigkeit besteht darin, die Temperatur zu bestimmen. Im Falle des 2D-Ising-Modells ist der Zusammenhang zwischen Energie und Temperatur exakt bekannt. Im allgemeinen Fall bestimmt man die Temperatur aus dem Histogramm der Energieverteilungen der Dämonen, welche ja nach Boltzmann die Form $e^{-E/kT}$ hat.

6.4.5 Kritisches slowing down, Clusteralgorithmus

Bestimmt man z.B. die Magnetisierung des Ising-Modells als Funktion der Monte-Carlo-Steps pro Gitterplatz (das entspricht der Zeit), dann erfolgt die Relaxation auf einen konstanten Gleichgewichtswert M_s gemäß $(M - M_s) \propto \exp(-t/\tau)$ (siehe Kapitel

1). Die Relaxationszeit τ selber folgt in der Nähe einer kritischen Temperatur T_c dem Gesetz

$$\tau \propto |T - T_c|^{-z}$$

z ist der dynamische kritische Exponent. Für $T \approx T_c$ erfolgt die Relaxation ins Gleichgewicht also sehr langsam. Diesen Effekt nennt man *critical slowing down*. Ein zweidimensionales Ising-Modell hat als dynamischen kritischen Exponenten $z \approx 2.1$ (Metropolis, Glauber oder heat-bath).
Die Simulation einer binären Mischung hat als dynamischen kritischen Exponenten $z \approx 4$ (Kawasaki). Das bedeutet, daß die Relaxation ins Gleichgewicht hier noch langsamer erfolgt. Das Modell einer binäre Mischung entspricht dem Ising-Modell, jedoch mit dem Unterschied, daß die Magnetisierung (hier: Teilchenzahl jeder Gassorte) erhalten bleibt.
Ein Modell, bei dem zusätzlich noch Energieerhaltung gilt (Creutz) hat einen dynamischen kritischen Exponenten, der sehr viel größer ist als 2.
Es soll nun ein Weg erläutert werden, mit dem man das critical slowing down umgehen kann. Man betrachte dazu wieder ein Ising-Modell. Die Energie berechnet sich nach

$$E = J\sum_{v}\varepsilon_v \quad \text{mit} \quad \varepsilon_v = \begin{cases} 0 & \text{falls Endpunkte gleich sind} \\ 1 & \text{falls Endpunkte verschieden sind} \end{cases}$$

$\sum_v$ soll über alle Verbindungen im Gitter summieren. Man definiert nun die beiden Operationen **C** und **D** (siehe Abbildung). **C** kontrahiert zwei Gitterpunkte zu einem Gitterpunkt. **D** entfernt die Verbindung zwischen zwei Gitterpunkten.

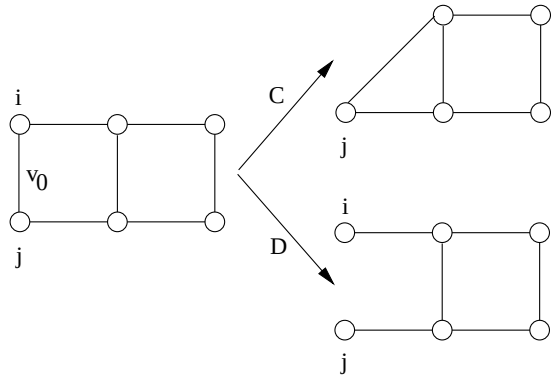


Abb. 6.3 *Wirkung der Operationen C und D*

Für die Zustandssumme gilt:

$$Z = \sum_{Konf.} \exp(-\frac{E}{kT}) = \sum_{Konf.} \prod_{v} exp(-\frac{J\varepsilon_v}{kT})$$

$\sum_{Konf.}$ summiert über alle möglichen Konfigurationen. Im zweiten Schritt wurde obige Definition der Energie eingesetzt. Es soll nun die Verbindung v_0 zwischen i und j

betrachtet werden. Zieht man ihren Energiebeitrag aus der Summe heraus, erhält man:

$$\begin{aligned} Z &= \sum_{Konf.} \exp(-\frac{J\varepsilon_{v_0}}{kT}) \prod_{v\neq v_0} \exp(-\frac{J\varepsilon_v}{kT}) \\ &= \sum_{\substack{\text{Konf.} \\ s_i=s_j}} \prod_{v\neq v_0} \exp(-\frac{J\varepsilon_v}{kT}) + \exp(-\frac{J}{kT}) \sum_{\substack{\text{Konf.} \\ s_i\neq s_j}} \prod_{v\neq v_0} \exp(-\frac{J\varepsilon_v}{kT}) \end{aligned}$$

Bei der zweiten Umformung wurde die Summe über die Konfigurationen aufgeteilt: Der erste Term enthält alle Konfigurationen, in denen die Zustände auf den Plätzen i und j gleich sind. Der zweite Term enthält alle Terme mit unterschiedlichem Zuständen auf i und j. ε_{v_o} wurde gleichzeitig entsprechend seiner Definition ersetzt. Der erste Term ist nun aber nichts anderes, als die Zustandssumme Z_C des ursprünglichen Gitters nach der Anwendung des Operators **C** auf die Gitterplätze i und j. Z_D entspricht der ursprünglichen Zustandssumme ohne den Beitrag der Verbindung v_0, enthält jedoch die Beiträge für beliebige i und j. Zur Darstellung des zweiten Terms, muß man also noch den Beitrag Z_C der Konfigurationen mit $s_i = s_j$ abziehen. Die Zustandssumme erhält damit die Darstellung:

$$\begin{aligned} Z &= Z_C + \exp\left(-\frac{J}{kT}\right)(Z_D - Z_C) \\ &= \left(1 - \exp\left(-\frac{J}{kT}\right)\right) Z_C + \exp\left(-\frac{J}{kT}\right) Z_D \end{aligned}$$

Man kann nun diese beiden Operatoren so lange auf das Gitter anwenden, bis alle Verbindungen verschwunden sind und nur noch einzelne, unverbundene Gitterpunkte da sind. Das Ergebnis, das man dann erhält, ist das gleiche, wie wenn man bei einer Perkolationskonfiguration alle Cluster jeweils auf einen Punkt zusammenschrumpft. Bei ursprünglich M Verbindungen erhält man 2^M verschiedene Endzustände. Für die Zustandssumme kann man dann schreiben:

$$Z = \sum_{\substack{\text{Perkolations-} \\ \text{konfiguration}}} 2^{Cluster} (1 - \exp(-\frac{J}{kT}))^c \exp(-\frac{J}{kT})^d$$

Die Exponenten sind die Clusteranzahl, die Anzahl c von **C**-Operationen und die Anzahl d von **D**-Operationen. Auf der rechten Seite berücksichtigt der erste Term $\sum_{\substack{\text{Perkolations-} \\ \text{konfiguration}}} 2^{Cluster}$ die Möglichkeit, daß jedes Cluster sowohl aus Einsen als auch aus Nullen gebildet werden kann. Die restlichen beiden Terme entsprechen gerade dem Gewicht einer Perkolationskonfiguration:

$$< \cdots >= \sum_{Perkolationen} p^{besetzt} (1-p)^{unbesetzt},$$

also hat man,

$$Z = < 2^{Cluster} > .$$

Die Exponenten sind die Anzahl der besetzten Verbindungen und die Anzahl der unbesetzten Verbindungen. In unserem Fall wählt man die Besetzungswahrscheinlichkeit als $p = 1 - \exp(-\frac{J}{kT})$. Diese Darstellung geht auf Kasteleyn und Fortuin (1969) zurück.
Der Swendsen Wang Algorithmus (1987) zur Vermeidung des critical slowing down geht nun folgendermaßen vor:

1. Mit der Wahrscheinlichkeit p besetzt man alle Verbindungen mit gleichen Endpunkten. (Ungleiche Endpunkte werden nie verbunden.) Auf diese Weise entstehen Cluster.
2. Nun wählt man ein oder mehrere Cluster aus und invertiert die Zustände *aller* Gitterpunkte im Cluster.
3. Starte wieder mit dem 1.Schritt.

Man sieht leicht, daß die Übergangswahrscheinlichkeit zwischen zwei Konfigurationen gerade im Unterschied der Anzahl der mit Wahrscheinlichkeit $1-p$ nicht besetzten Verbindungen mit gleichem Spin liegt und sonst die klassische Bedingung des detailed balance erfüllt ist.
Auch der *Wolff*-Algorithmus (1988) kehrt die Zustände eines ganzen Clusters um. Es wird dazu jedoch nur ein Cluster generiert, das später dann auch invertiert wird. Die Idee zur Erzeugung eines Clusters stammt von *Leath*:

1. Man startet bei einem beliebigen Punkt im Gitter (z.B. einem Spinzustand $+1$). Alle nächsten Nachbarn dieses Punktes mit Spinzustand $+1$ sind sogenannte Wachstumsplätze.
2. Man erzeugt eine Zufallszahl z. Ist $z < p$ (hier: $p = 1 - \exp(-\frac{J}{kT})$), wird ein Wachstumsplatz besetzt. Ist $z > p$ wird der Wachstumsplatz nicht besetzt und für immer verboten. D.h. er kann auch im weiteren Programmablauf nicht mehr besetzt werden.
3. Die nächsten Nachbarn eines neu besetzten Platzes mit Spinzustand $+1$ sind neue, zusätzliche Wachstumsplätze.
4. der Algorithmus wird solange fortgeführt, bis das ganze Cluster nur noch verbotene Plätze als nächste Nachbarn hat.

Diese beiden Algorithmen erreichen kritische Exponenten von $z \approx 0.3$ in zwei Dimensionen und $z \approx 0.55$ in drei Dimensionen.

6.5 Histogrammmethoden

Hat man Ergebnisse einer Simulation bei der Temperatur T, möchte aber Ergebnissse für die Temperatur T^*, dann ist es sehr unökonomisch, eine erneute Simulation bei der neuen Temperatur T^* durchzuführen. Man kann stattdessen die Ergebnisse mit einer *Histogrammethode* umrechnen.
Es soll zunächst die von Salzburg et al. (1959), sowie von Ferrenberg und Swendsen (1989) untersuchte Methode vorgestellt werden: Man möchte den Erwartungswert

einer Größe Q_T berechnen.

$$Q_T = \frac{1}{Z_T} \sum_E Q(E) p_T(E)$$

mit $Z_T = \sum_E p_T(E)$. Man beachte, daß $p_T(E)$ hier die Wahrscheinlichkeit dafür ist, daß das System die Energie E hat, nicht die Wahrscheinlichkeit, daß es den Zustand der Energie E hat.

$$p_T(E) = g(E) \exp(-\frac{E}{kT})$$

$g(E)$ ist die Anzahl der Zustände mit der Energie E. Bei einer anderen Temperatur T^* gilt:

$$Q_{T^*} = \frac{1}{Z_{T^*}} \sum_E Q(E) p_{T^*}(E)$$

Für die Wahrscheinlichkeit $p_{T^*}(E)$ gilt jetzt:

$$p_{T^*}(E) = g(E) \exp(-\frac{E}{kT^*}) = p_T(E) \exp(-\frac{E}{kT^*} + \frac{E}{kT})$$

Der Umrechnungsfaktor $\exp(-\frac{E}{kT^*} + \frac{E}{kT})$ wird im folgenden als f_{TT^*} abgekürzt. Ohne daß dazu eine neue Simulation nötig wäre, erhält man den neuen Erwartungswert Q_{T^*} bei der Temperatur T^* aus

$$Q_{T^*} = \frac{\sum_E Q(E) p_T(E) f_{TT^*}}{\sum_E p_T(E) f_{TT^*}}$$

Diese Methode hat ein Problem: Die in der Simulation z.B. mit der Methode von Metropolis erzeugten Konfigurationen (aus denen wiederum die Werte von $Q(E)$ bestimmt wurden) sind im Konfigurationsraum nicht gleichverteilt. Es werden bevorzugt solche Systemzustände erzeugt, deren Wahrscheinlichkeiten im Bereich des Maximums von $p_T(E)$ liegen. Ändert sich die Temperatur des Systems, ändert sich auch die mittlere Energie des Systems und damit die Lage des Maximum von $p_T(E)$. Da das Maximum zudem äußerst scharf ist, ist der Überlapp von $p_T(E)$ und $p_{T^*}(E)$ sehr klein. Das bedeutet, daß nur sehr wenige Konfigurationen mit Wahrscheinlichkeiten im Bereich des Maximums von $p_{T^*}(E)$ vorhanden sind. Man hat daher eine sehr schlechte Statistik. Das Problem wird umso schlimmer, je größer $|T - T^*|$ und je größer das System ist.

Eine Methode zur Umgehung dieses Problems ist die *Breite Histogramm Monte Carlo* Methode (BHMC 1996). Sie verwendet einen Markovprozeß im Energieraum, der so gewählt wird, daß das ganze Energieintervall gleichmäßig abgetastet wird. Man führt zunächst zwei Variablen ein: N_{up} ist die Anzahl aller Prozesse, bei denen eine Energieerhöhung $E \to E + \Delta E$ stattfindet. N_{down} ist die Anzahl aller Prozesse, bei denen eine Energieerniedrigung $E \to E - \Delta E$ stattfindet. Es soll wieder eine Gleichgewichtsbedingung wie die des detailed balance gelten:

$$g(E + \Delta E) N_{down}(E + \Delta E) = g(E) N_{up}(E)$$

Logarithmiert man diese Gleichung, erhält man:

$$\log g(E+\Delta E) - \log g(E) = \log N_{up}(E) - \log N_{down}(E+\Delta E)$$

Multipliziert man diese Gleichung mit $1/\Delta E$, steht auf der linken Seite gerade der Differenzenquotient von $\log g(E)$. Für kleine ΔE kann man dann schreiben:

$$\frac{\partial \log g(E)}{\partial E} = \frac{1}{\Delta E} \log \frac{N_{up}(E)}{N_{down}(E+\Delta E)}$$

Man hat nun folgendes Vorgehen:

1. Man nimmt eine Konfiguration und prüft für jeden Gitterplatz, ob eine Änderung des Zustandes die Energie erniedrigt oder erhöht. Im ersten Fall erhöht man N_{down} um eins, im zweiten Fall erhöht man N_{up} um eins.
2. Dann wählt man zufällig einen Gitterplatz. Die Änderung an diesem Gitterplatz akzeptiert man, falls die Energie dabei erniedrigt wird. Wird die Energie erhöht, akzeptiert man die Änderung nur mit der Wahrscheinlichkeit N_{down}/N_{up}.
3. Aus der Simulation bestimmt man $N_{down}(E)$, $N_{up}(E)$ und $Q(E)$.

Aus den Werten von $N_{down}(E)$ und $N_{up}(E)$ kann man anschließend $g(E)$ aus obiger Formel berechnen. Der Erwartungswert von Q_T bei einer beliebigen Temperatur ergibt sich dann aus:

$$Q_T = \frac{\sum_E Q(E) g(E) \exp(-\frac{E}{kT})}{\sum_E g(E) \exp(-\frac{E}{kT})}$$

Die Ergebnisse sollen nun an einem eindimensionalen Ising-Modell mit N Plätzen überprüft werden: Zwei benachbarte Plätze im gleichen Zustand sollen keinen Beitrag zur Energie leisten. Zwei benachbarte Plätze in verschiedenen Zuständen erhöhen die Energie um 1. Für die Anzahl der Zustände gilt:

Gitterplätze	$g(E)$			
	$E=0$	$E=1$	$E=2$	$E=3$
1	2			
2	2	2		
3	2	4	2	
4	2	6	6	2

Für N Gitterplätze gilt:

$$g(E) = 2\frac{N!}{E!(N-E)!}$$

Mit der Stirlingschen Näherungsformel $\ln x! = x\ln x - x$ berechnet man:

$$\begin{aligned} \frac{\partial \log g(E)}{\partial E} &= \frac{\partial}{\partial E} \log\left(2\frac{N!}{E!(N-E)!}\right) \\ &= \frac{\partial}{\partial E}(\log 2 + \log N! - \log E! - \log(N-E)!) \end{aligned}$$

$$
\begin{aligned}
&= \frac{\partial}{\partial E}(-E\log E + E - (N-E)\log(N-E) + (N-E)) \\
&= -\log E + \log(N-E) \\
&= \log\left(\frac{N-E}{E}\right)
\end{aligned}
$$

Betrachtet man alle Verbindungen zwischen Plätzen in verschiedenen Zuständen, dann ist die Anzahl der Verbindungen gleich E. Um die Energie zu erniedrigen, muß man eine Verbindung entfernen, d.h. $N_{down} = E$. Entsprechend ergeben sich $N_{up} = N - E$ Möglichkeiten, noch eine Verbindung hinzuzufügen. Das entspricht genau dem oben erhaltenen Ergebnis.

Kapitel 7

Die Monte-Carlo-Methode Teil II

7.1 Lösung von Integralen mittels MC

Ein bekanntes Beispiel zur Lösung von Integralen mittels MC berechnet die Zahl π. Dazu wählt man zunächst zwei auf dem Intervall $[0,1]$ gleichverteilte Zufallszahlen x und y. Liegt der Punkt $P(x,y)$ innerhalb der Fläche des Viertelkreises (siehe Abbildung), dann erhöht man einen Zähler c um eins. Dieses Verfahren wiederholt man für N Paare von Zufallszahlen. Für π erhält man so:

$$\pi(N) = 4\frac{c}{N}$$

Für den Fehler des so berechneten π gilt:

$$\Delta = |\pi - \pi(N)| \propto \frac{1}{\sqrt{N}}$$

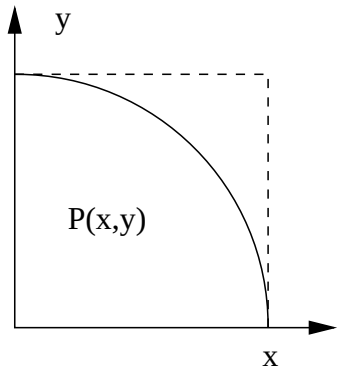


Abb. 7.1 *Bestimmung der Zahl π mit der Monte-Carlo-Methode*

Es soll nun die allgemeine Methode vorgestellt werden. Dazu sei zunächst an die Berechnung des Erwartungswertes EV einer Verteilung $p(x)$ erinnert (siehe Kapitel 1).

$$\mathrm{EV}(x) = \int dx\, xp(x) \approx \frac{1}{N}\sum_{i=1}^{N} x_i.$$

Analog gilt für den Erwartungswert einer Funktion $g(x)$

$$\mathrm{EV}(g(x)) = \int dx\, g(x)p(x) \approx \frac{1}{N}\sum_{i=1}^{N} g(x_i).$$

x_i sind jeweils die mit der Verteilung $p(x)$ erzeugten Zufallszahlen.
Wählt man für $p(x)$ die Gleichverteilung

$$p(x) = p_0(x) = \frac{1}{b-a} \quad \text{für} \quad x \in [a,b] \quad \text{und } 0 \quad \text{sonst},$$

so sieht man, daß sich mittels Zufallszahlen Integrale auswerten lassen:

$$\begin{aligned} \frac{1}{b-a}\int_a^b dx g(x) &\approx \frac{1}{N}\sum_{i=1}^{N} g(x_i) \\ \Leftrightarrow \int_a^b dx g(x) &\approx \frac{b-a}{N}\sum_{i=1}^{N} g(x_i) \end{aligned}$$

Da die Abtastung, das *sampling*, mit gleichverteilten Zufallszahlen erfolgte, spricht man von einem *simple sampling*. Dieses Verfahren funktioniert gut für Integranden, die genügend "glatt" und auf begrenzten Intervallen definiert sind.
Bei stark variablen Integranden oder Funktionen, die sich bis ins Unendliche erstrecken, funktioniert dieser naive Zugang nicht, denn nur sehr wenige der Zufallszahlen werden in den Bereich fallen, in dem die Funktion wesentlich von 0 verschieden ist (bzw. große Werte annimmt). Der Mittelwert der rechten Seite der Gleichung wird daher sehr starken Schwankungen unterworfen sein, und man muß unpraktikabel lange Serien von Zufallszahlen auswerten.
Dieses Verhalten läßt sich verbessern, wenn man andere Verteilungen als die Gleichverteilung zuläßt,

$$\int dx\, g(x) = \int dx\, p(x)\frac{g(x)}{p(x)} \approx \frac{1}{N}\sum_{i=1}^{N}\frac{g(x_i)}{p(x_i)},$$

wobei das sampling (rechte Seite) jetzt natürlich bezüglich der Verteilung $p(x)$ zu verstehen ist. Der Mittelwert konvergiert am schnellsten, wenn die Funktion $g(x)/p(x)$ möglichst "glatt," also konstant ist. Leider bedeutet das, daß die Verteilung der Zufallszahlen zu $g(x)$ proportional sein sollte. "Leider" deswegen, weil die Erzeugung von Zufallszahlen einer beliebigen Dichte i.a. die Integration der Verteilung, hier $g(x)$, und die Bildung der Umkehrfunktion beinhaltet. Das Problem hat durch diese Form des sogenannten *importance samplings* also eher an Komplexität zugenommen.
Es gibt aber zwei Auswege aus diesem Dilemma,

- man kann $p(x)$ so wählen, daß die Proportionalität nur approximativ realisiert wird, z.B. durch eine abschnittsweise konstante Funktion, eine angepaßte Gaußverteilung, etc., so daß sich die Zufallszahlen in einfacher Weise generieren lassen. (siehe Kapitel 2)

- oder man kann ganz vom Konzept unabhängiger Zufallszahlen abgehen, und sich stattdessen eine korrelierte Folge x_i beschaffen, die asymptotisch die richtige Verteilung hat. D.h. man macht einen Markov-Prozeß, wie ihn z.B. das Metropolis-Verfahren verwendet, wobei $p(x)$ hier nicht mehr die Boltzmann-Verteilung sein muß, sondern entsprechend $p(x) \propto g(x)$ gewählt wird.

Man verwendet zur numerischen Berechnung eines Integrales $\int_a^b f(x)dx$ oft die "konventionelle" Methode, d.h. man unterteilt das Intervall $[a,b]$ in äquidistante Stützstellen im Abstand h, und berechnet $h\sum_{i=1}^{N} f(x_i)$. Dann gilt für den Fehler dieser eindimensionalen Integration:

$$\Delta \propto h^2 \propto \frac{1}{N^2}$$

Hier wurde eingesetzt, daß für die Anzahl N der Stützstellen bzw. der Rechenschritte $N \propto \frac{1}{h}$ gilt.

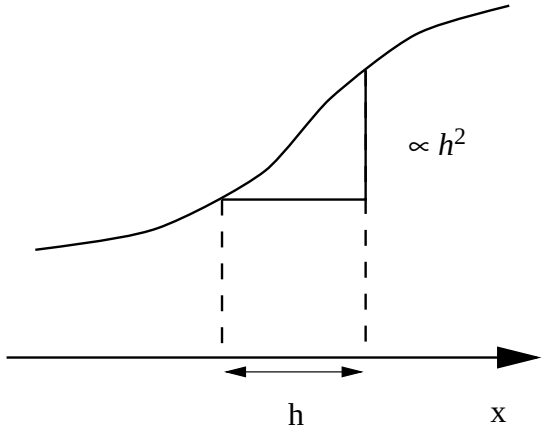


Abb. 7.2 *Fehlerabschätzung bei einer eindimensionalen Integration*

Bei einer d-dimensionalen Integration summiert man nicht mehr über Intervalle der Länge h, sondern über "Quader" mit dem Volumen h^d. Für die Anzahl der Rechenschritte gilt jetzt $N \propto \frac{1}{h^d}$. Entsprechend gilt für den Fehler:

$$\Delta \propto h^2 \propto N^{-\frac{2}{d}}$$

Der Fehler der Monte-Carlo-Methode hingegen wächst auch bei höheren Dimensionen nur mit $\Delta \propto 1/\sqrt{N}$. Das bedeutet, daß bereits für Dimensionen $d > 4$ die Monte-Carlo-Methode mit zunehmendem Rechenaufwand wesentlich schneller konvergiert als die konventionelle Methode.

Als Beispiel sei nun ein zweidimensionales System von n gleichen, klassischen Teilchen in einem quadratischen Kasten mit Kantenlänge L gewählt. Der Teilchendurchmesser sei s. Zwischen den Teilchen wirke ein Hard-Core-Potential. Gesucht ist der mittlere Teilchenabstand (Mittelung über den Phasenraum)

$$< r >= \frac{1}{Z} \int \sum_{i<j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\, d\vec{X}$$

Z ist das Phasenraumvolumen. Als Nebenbedingungen hat man

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} > s.$$

Der Phasenraum ist $2n$-dimensional. Entsprechend besitzt auch das Integral diese Dimension. Da man zudem noch Nebenbedingungen hat, ist die Monte-Carlo-Methode hier die einzige Methode, die eine Lösung dieses Integrals ermöglicht. Die Auswertung des Integrals umfaßt folgende Schritte:

1. Man wählt ein beliebiges Teilchen i aus.
2. Man bestimmt zwei Zufallszahlen $x_i, y_i \in [0, L]$ und setzt das Teilchen i "probeweise" an die Position (x_i, y_i). Erfüllt die neue Position die obigen Nebenbedingungen für alle j (d.h. das Teilchen i überlappt kein anderes Teilchen), beläßt man das Teilchen in dieser Position. Im anderen Fall wird es wieder zurückgesetzt und Schritt 1 wiederholt.
3. In der so gewonnen Konfiguration berechnet man nun

$$< r >= \sum_{i<j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Das ganze Verfahren wird solange wiederholt, bis der 3. Schritt N-mal durchgeführt wurde. Das Mittel über alle dabei bestimmten $< r_{ij} >$ liefert dann den mittleren Teilchenabstand mit dem Fehler $\Delta \propto 1/\sqrt{N}$.
Möchte man ein realistischeres Potential $V(r)$ berücksichtigen, akzeptiert man im 2. Schritt die neue Teilchenposition zusätzlich nur mit der Akzeptierwahrscheinlichkeit $A = \min(1, \exp(-\frac{E^* - E}{kT}))$, wobei sich die Energie E einer Konfiguration aus dem vorgegebenen Potential $V(r)$ errechnet.

7.2 Quanten-Monte-Carlo

7.2.1 Variationelles Monte-Carlo-Verfahren

Bei der Behandlung quantenmechanischer Systeme möchte man Größen der Form

$$\langle \Psi | A | \Psi \rangle = \frac{\int d\vec{X} \Psi^*(\vec{X}) A \Psi(\vec{X})}{\int d\vec{X} \Psi^*(\vec{X}) \Psi(\vec{X})}$$

berechnen. Wobei zunächst nur bosonische Systeme betrachtet werden sollen. Das Problem dabei ist, daß

1. die Funktion Ψ unbekannt ist.
2. die Übergangswahrscheinlichkeit nicht mehr durch den Boltzmannfaktor gegeben ist, sondern sich aus $|\Psi|^2$ berechnet.

Man wählt nun für Ψ eine Testfunktion

$$\Psi_T(\vec{X}) = \exp(\frac{1}{2} \sum_{i<j} u_{a_k}(r_{ij})) \qquad \text{Jasnow (1958)}$$

$\vec{X}$ ist der Konfigurationsvektor $\vec{X} = (\vec{r}_1, \ldots, \vec{r}_N)$, wobei $\vec{r}_i$ der Ortsvektor des i-ten Teilchens ist. u ist ein Pseudopotential, das nur vom Teilchenabstand und den Parametern a_k abhängen soll. Eine mögliche Wahl für u ist z.B. das Lennard-Jones-Potential

$$u(r) = 4\varepsilon \left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right)$$

mit den Parametern ε und σ.

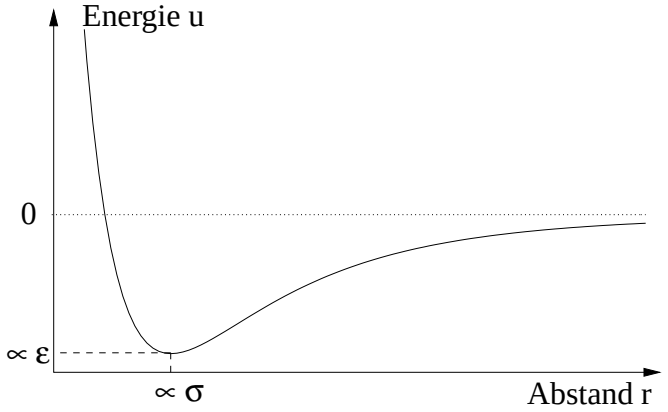


Abb. 7.3 *Lennard-Jones-Potential*

Die Parameter werden nun so lange variiert, bis die Energie

$$E_T = \langle T|H|T\rangle$$

mit dem System-Hamiltonoperator H minimal wird. Zur Auswertung dieses Integrals wendet man selbst wieder ein Monte-Carlo-Verfahren an. Hat man so schließlich die Parameter a_k bestimmt, kann man damit die Erwartungswerte aller weiteren Operatoren A mit dem Monte-Carlo-Verfahren berechnen.
Es ergibt sich somit folgendes Vorgehen:

1. Wähle einen Satz von Parametern a_k.
2. Wähle eine Anfangskonfiguration $\vec{X}$.
3. Versetze ein beliebiges Teilchen i probeweise von $r_i \to \vec{r_i'}$.
4. Akzeptiere diesen Zug mit der Wahrscheinlichkeit $A = \min(1, p_B)$ mit

$$p_B = \frac{|\Psi_T(\vec{X}')|^2}{|\Psi_T(\vec{X})|^2} = \exp(\sum_j (u(\vec{r_i'} - \vec{r_j}) - u(\vec{r_i} - \vec{r_j})))$$

   (falls man die Testfunktion wieder nach Jasnow wählt.).
5. Wiederhole Schritt 3 und 4 bis M neue Konfigurationen generiert wurden.
6. Berechne

$$E_T = \frac{\sum_j^M \Psi^*(\vec{X}_j) H(\vec{X}_j) \Psi(\vec{X}_j)}{\sum_j^M \Psi^*(\vec{X}_j) \Psi(\vec{X}_j)}$$

7. Ist das neu berechnete E_T kleiner als das im vorigen Durchgang berechnete, übernimmt man die neuen Paramter a_k.
8. Man wählt einen neuen Satz von Parametern a_k (z.B. durch ein Newton-Verfahren) und springt zu Schritt 2.

Nach dem Durchlaufen dieses Algorithmus besitzt man einen Satz von a_k, für den die Energie minimal ist. Damit kann man nun die weiteren Rechnungen durchführen.
Als Testfunktion zur Behandlung von Fermionen wählt man

$$\Psi_T^F = det(D)\Psi_T$$

$det(D)$ ist die Slaterdeterminante. Für D wählt man

$$D_{jk} = \exp(i\vec{k}\vec{r}_j)$$

Die Wahrscheinlichkeit p_F von Fermionen ergibt sich dann aus der obigen Wahrscheinlichkeit entsprechend

$$p_F = |\frac{det(D')}{det(D)}|^2 p_B = \sum_k \exp(i\vec{k}(\vec{r}'_j - \vec{r}_j))\bar{D}_{jk} p_B$$

$\bar{D}$ ist die inverse Matrix von D. $\sum_k D_{ik}\bar{D}_{jk} = \delta_{ij}$. Damit läßt sich die Behandlung von Fermionen auf die Behandlung von Bosonen zurückführen.

7.2.2 Greensfunktion Monte-Carlo

Gegeben sei der Hamiltonoperator H mit den Eigenwerten $E_{max} > E_i > E_{min}$. Es soll nun der Operator der Greensfunktion definiert werden durch

$$G = I - \tau(H - \omega I)$$

τ und ω sind Konstanten. I ist der Identitätsoperator. G geht somit aus H durch eine Verschiebung hervor. Wählt man $\omega = E_{min} + \varepsilon$ mit $\varepsilon \ll 1$, gilt für die Eigenwerte λ_i von G:

$$-1 \leq \lambda_i \leq 1 + \tau\varepsilon$$

Das wird deutlich, wenn man E_{min} bzw. E_{max} einsetzt:

$$1 - \tau(E_{min} - \omega) = 1 - \tau(E_{min} - E_{min} - \varepsilon) = 1 + \tau\varepsilon = \lambda_{max}$$

$$1 - \tau(E_{max} - \omega) > 1 \quad \text{für} \quad \tau < \frac{2}{E_{max} - \omega}$$

Durch geeignete Wahl von τ und ω besitzt somit auch G ein beschränktes Spektrum, wobei dem größten Eigenwert von H der kleinste Eigenwert von G zugeordnet ist.

Ein Verfahren zur numerischen Bestimmung des größten Eigenwertes eines Operators stammt von Lanczos. Es seien $|\Phi_i\rangle$ die Eigenfunktionen des Hamiltonoperators mit den Eigenwerten E_i. G auf $|\Phi_i\rangle$ angewandt liefert dann

$$G|\Phi_i\rangle = (1-\tau(E_i-\omega))|\Phi_i\rangle = \lambda_i|\Phi_i\rangle$$

Hier wurde $\lambda_i = 1-\tau(E_i-\omega)$ gesetzt. Die Funktionen $|\Psi\rangle$ lassen sich in die $|\Phi_i\rangle$ entwickeln gemäß

$$|\Psi\rangle = \sum_i a_i|\Phi_i\rangle$$

G einmal auf $|\Psi\rangle$ angewandt liefert

$$G|\Psi\rangle = \sum_i \lambda_i a_i|\Phi_i\rangle$$

und die n-fache Anwendung

$$G^n|\Psi\rangle = \sum_i \lambda_i^n a_i|\Phi_i\rangle$$

Sind nun die Eigenwerte λ_i wie oben beschränkt, und gibt es einen größten Eigenwert λ_{max} mit $\lambda_{max} = \lambda_0 > \lambda_1 > \cdots > \lambda_{min}$, dann gilt $\lambda_{max}^n = \lambda_0^n \gg \lambda_1^n \gg \cdots \gg \lambda_{min}^n$. Die n-fache Anwendung von G auf $|\Psi\rangle$ kann man dann näherungsweise schreiben als:

$$G^n|\Psi\rangle \approx \lambda_0^n a_0|\Phi_0\rangle \approx \lambda_o G^{n-1}|\Psi\rangle$$

Mit

$$\lambda_0 = \lim_{n\to\infty}\frac{|G^n|\Phi_0\rangle|}{|G^{n-1}|\Phi_0\rangle|} \qquad \text{und} \qquad |\Phi_o\rangle = \frac{1}{a_0}\lim_{n\to\infty}\frac{|G^n|\Phi_0\rangle|}{\lambda_0^n}$$

kann man somit den größten Eigenwert und die zugehörige Eigenfunktion numerisch bestimmen.
Um nun G und $|\Psi\rangle$ tatsächlich zu bestimmen, definiert man zunächst die Vielteilchenfunktion

$$|X\rangle = \prod_{i=1}^{N} b_{\vec{r}_i}^+|0\rangle$$

$|0\rangle$ ist die Wellenfunktion des Vakuums. $b_{\vec{r}_i}^+$ erzeugt das Teilchen i am Ort $\vec{r}_i$. $|\Psi\rangle$ soll sich in der Basis der $|X\rangle$ darstellen lassen.

$$|\Psi\rangle = \sum_X \Psi(X)|X\rangle$$

$\Psi(X) = \langle X|\Psi\rangle$ ist die Wahrscheinlichkeitsdichte. Mit der Abkürzung $|\Psi_n\rangle = G^n|\Psi\rangle$ schreibt man nun:

$$\begin{aligned} |\Psi_n\rangle &= G|\Psi_{n-1}\rangle \\ \Leftrightarrow \langle X|\Psi_n\rangle &= \langle X|G|\Psi_{n-1}\rangle \end{aligned}$$

$$\begin{aligned} &= \sum_{X'} \langle X|G|X'\rangle\langle X'\Psi_{n-1}\rangle \\ \Leftrightarrow \Psi_n(X) &= \sum_{X'} G(X,X')\Psi_{n-1}(X') \end{aligned}$$

Für $G(X,X')$ muß damit gelten:

$$G(X,X') = \begin{cases} 1-\tau(u(X')-\omega) & \text{für } X'=X \\ V(X',X) & \text{falls Übergang } X' \to X \text{ möglich} \\ 0 & \text{sonst} \end{cases}$$

mit $u(X') = \langle\Phi_0|H|\Phi_0\rangle$.
$G(X,X')$ soll nun als Produkt einer Aufenthalts- und einer Übergangswahrscheinlichkeit geschrieben werden.

$$G(X,X') = P(X,X')W(X')$$

Die Wahrscheinlichkeitsdichte kann man damit schreiben als:

$$\begin{aligned} \Psi_n(X) &= \langle X|\Psi_n\rangle \\ &= \langle X|G^n|\Psi_0\rangle \\ &= \sum_{X_n\ldots X_0} \langle X|X_n\rangle\langle X_n|G|X_{n-1}\rangle\ldots\langle X_1|G|X_0\rangle\langle X_0|\Psi_0\rangle \\ &= \sum_{X_n\ldots X_0} \delta(x,x_n)W(X_n)\ldots W(X_1)\underbrace{P(X_n,X_{n-1})\ldots P(X_1,X_0)\Psi_0(X_0)} \end{aligned}$$

Der unterklammerte Term ist die Wahrscheinlichkeit dafür, in einem Markov-Prozeß, der im Zustand X_0 startet und mit der Wahrscheinlichkeit $P(X_{i-1},X_i)$ vom Zustand X_{i-1} zum Zustand X_i übergeht, die Kette $X_0 \to X_1 \to \cdots \to X_n$ zu bekommen. Erzeugt man mit einem solchen Prozeß k Ketten, dann haben die einzelnen Ketten automatisch das richtige Gewicht, und die Mittelung

$$\Psi_n(X) = \frac{1}{k}\sum_k \delta(X-X_n)\prod_{j=1}^{n} W(X_j)$$

über alle erzeugten Ketten liefert $\Psi_n(X)$.

7.3 Monte-Carlo am kritischen Punkt

Die Eigenschaften eines Systems am kritischen Punkt sind charakterisiert durch:
1. Die *Divergenz charakteristischer Zeiten und/oder Längen,* die sich ergibt aus
2. dem *Potenzgesetzverhalten* für große Abstände r von Korrelationsfunktionen wie z.B. der Zweipunktkorrelationsfunktion der Magnetisierung $\langle M(0)M(r)\rangle - \langle M(0)\rangle^2$, die sonst für große Abstände ein exponentiell abklingendes Verhalten zeigen. Es

läßt sich zeigen, daß die Suszeptibilität als Integral über diese Korrelationsfunktion geschrieben werden kann

$$\chi = \frac{1}{kT} \int d^3r \ \left(\langle M(0)M(r) \rangle - \langle M(0) \rangle^2 \right)$$

und daß dieses Integral am kritischen Punkt aufgrund eines zu langsam abfallenden Potenzgesetzverhaltens des Integranden divergiert.

3. Potenzgesetze am kritischen Punkt ergeben sich als Folge der sogenannten *Skaleninvarianz* des Systems — dabei bleiben die Eigenschaften des Systems (z.B. die Größenverteilung der Bereiche mit gleichorientierten Spins) *unabhängig von der Längenskala*, auf der man das System beschreibt. Im Gegensatz dazu sieht beispielsweise kochendes Wasser auf verschiedenen Skalen sehr wohl verschieden aus, denn wenn man sich weit genug wegbewegt, also auf großen Sklaen beobachtet, dann sieht man keine Dampfbläschen mehr.

Die Skaleninvarianz ist der Grundgedanke der Renormierungsgruppe, die eine sehr leistungsfähige Näherungsmethode zur Beschreibung des Systemsverhaltens am kritischen Punkt ist. Dabei führt man eine Abbildung der physikalischen Beschreibung (d.h. der Zustandssumme mit bestimmten Werten für Temperatur und Magnetfeld) von einem “kleinskaligen” System auf ein “grobskaliges” System ein und fordert, daß die grundsätzliche Form der physikalischen Beschreibung erhalten bleiben muß (allerdings jetzt im allgemeinen mit einer neuen Temperatur und einem neuen Magnetfeld). Der kritische Punkt zeichnet sich dadurch aus, daß er ein Fixpunkt dieser Beschreibung ist, d.h. in unserem Beispiel, daß sich Temperatur und Magnetfeld unter Anwendung der Renormierungsabbildung auf eine um einen Skalenfaktor l vergrößerte Skala nicht ändern.

Die Theorie der Renormierung stammt von Wilson und Kadanoff (Block-Spins) (1966).

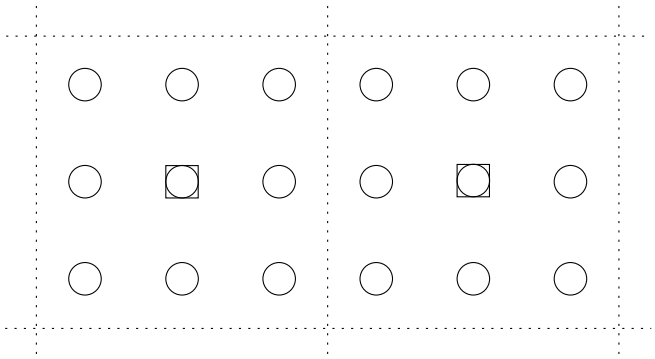


Abb. 7.4 *Renormierung eines Spinsystems. Die quadratischen Gitterplätze sind die Gitterplätze des renormierten Systems. Das Gitter des renormierten Systems ist gepunktet eingetragen.*

Als Beispiel soll ein Spinsystem auf einem quadratischen Gitter betrachtet werden. Das Gitter wird wie in der Abbildung gezeigt auf ein neues Gitter renormiert. Ein neuer Spinzustand $\tilde{s_j}$ ersetzt jetzt 9 alte Spinzustände s_i. Man sucht nun

a) die mathematische Formulierung der Transformation.
b) die Fixpunkte der Transformation.

Die freie Energie ist eine extensive Größe, d.h. sie ist proportional zur Anzahl der Freiheitsgrade im System. Diese Eigenschaft soll bei der Transformation erhalten bleiben:

$$\tilde{F}(\tilde{\varepsilon},\tilde{H}) = l^d F(\varepsilon,H).$$

Hier bedeutet $\varepsilon \equiv T - T_c$ der Abstand der Temperatur T von der kritischen Temperatur T_c, H ist ein äußeres Feld, l ist der Skalenfaktor und d ist die Dimension. In unserem Beispiel eines Spingitters ist $l = 3$ und $d = 2$.
In der Nähe eines kritischen Punkt gelten für sämtliche thermodynamische Größen sogenannte homogene Skalengesetze der folgenden Form (was sich aus der Linearisierung der Renormierungsabbildung (siehe unten) in der Nähe des den kritischen Punkt charakterisierenden Fixpunktes ergibt [1,2]), insbesondere also für die freie Energiedichte:

$$F(\varepsilon,H) = l^{-d} F(l^{y_T}\varepsilon, l^{y_H}H).$$

Der Vergleich mit der Formel darüber liefert.

$$\tilde{F}(\tilde{\varepsilon},\tilde{H}) = F(l^{y_T}\varepsilon, l^{y_H}H) \qquad \tilde{\varepsilon} = l^{y_T}\varepsilon, \qquad \tilde{H} = l^{y_H}H.$$

Der kritische Exponent ν charakterisiert die Divergenz der Korrelationslänge ξ mit der Annäherung an den kritischen Punkt $\xi \sim |\varepsilon|^{-\nu}$. Wenn man die Skalenänderung l auf die Änderung der Korrelationslänge $\varepsilon^{-\nu} = \xi$, unter Reskalierung $\tilde{\varepsilon}^{-\nu} = \tilde{\xi} = (\xi/l)$ anwendet, dann findet man, daß gilt $y_T = \frac{1}{\nu}$. Für y_H lassen sich ähnliche Beziehungen zu kritischen Exponenten finden.
Als Beispiel für die approximative Bestimmung einer Renormierungsabbildung soll die Dezimation betrachtet werden. Man erhält das renormierte System durch das "Streichen" jedes zweiten Spins. Für l gilt dann $l = \sqrt{2}$,.

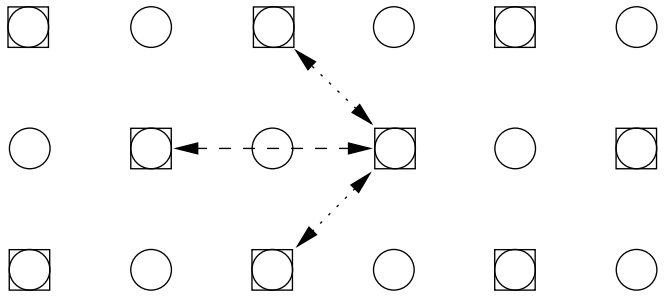


Abb. 7.5 *Die quadratischen Gitterplätze gehören zum renormierten System. Wechselwirkungen mit übernächsten Nachbarn (L) sind gestrichelt eingetragen. Wechselwirkungen mit nächsten Nachbarn (K) sind gepunktet dargestellt*

Es wird die Renormierung $R(\mathcal{H})$ der Hamiltonfunktion

$$\mathcal{H} = K \sum_{i,j=nn(i)} s_i s_j \qquad \text{mit } K = \frac{J}{kT}$$

betrachtet. Hier und im folgenden haben wir den Faktor $\frac{-1}{kT}$, der bei der Bildung der Zustandssumme im Exponenten der Exponentialfunktion auftaucht, in die Hamiltonfunktion absorbiert. Wir untersuchen dazu, was passiert, wenn wir die Beiträge eines zu eliminierenden Spins s_i in der ursprünglichen Zustandssumme explizit aufsummieren, hier bis auf Faktoren, die nicht von s_i abhängen:

$$\sum_{s_i=\pm 1} \exp(K \sum_{\tilde{m}(i)} s_i s_j) = 2\cosh(K \sum_{\tilde{m}(i)} s_j)$$

Die Summe $\sum_{\tilde{m}(i)}$ geht über die nächsten Nachbarn von s_i, die alle zum renormierten Gitter gehören. Der obige Beitrag zur Zustandssumme muß natürlich identisch sein mit einem entsprechenden Faktor in der Zustandssumme des renormierten Systems mit der Hamiltonfunktion $\tilde{\mathcal{H}}(\tilde{s})$. Dieser wird Beiträge aller der vom ursprünglichen Spin s_i "gekoppelten" Nachbarspins enthalten und wir wollen ihn in der folgenden Form

$$\exp(\tilde{\mathcal{H}}(\tilde{s})) = \exp(A(K) + B(K) \sum_2 \tilde{s}_i \tilde{s}_j + C(K) \sum_4 \tilde{s}_i \tilde{s}_j \tilde{s}_l \tilde{s}_m)$$

ansetzen. Die erste Summe umfaßt dabei die Spin-Spin-Wechselwirkungen über nächste und übernächste Nachbarn in unserer zu renormierenden Plakette. Die zweite Summe erfaßt neuartige Wechselwirkungen zwischen vier Spins, die im ursprünglichen $H(s)$ in dieser Form nicht vorhanden waren. Man kann nun folgende Fallunterscheidungen machen, um durch Vergleich der rechten Seiten unserer beiden letzten Gleichungen die Funktionen $A(K)$, $B(K)$, und $C(K)$ zu ermitteln:

1. Alle Spins sind im Zustand $+1$. Die Formel erhält dann die Gestalt:

$$2\cosh 4K = \exp(A + 6B + C)$$

2. Drei der vier benachbarten Spins sind im Zustand $+1$. Ein Spin ist im Zustand -1. Die Gleichung vereinfacht sich dann zu:

$$2\cosh 2K = \exp(A - C)$$

3. Zwei Spins sind im Zustand $+1$ und zwei im Zustand -1. Man erhält:

$$2 = \exp(A - 2B + C)$$

Aus diesen drei Gleichungen kann man die Koeffizienten A, B und C bestimmen. Als Lösung erhält man:

$$\begin{aligned} A &= \frac{1}{2}(\ln(4\cosh(2K)) + \frac{1}{4}\ln\cosh(4K)) \\ B &= \frac{1}{8}\ln\cosh(4K) \\ C &= \frac{1}{2}(\frac{1}{4}\ln\cosh(4K) - \ln\cosh(2K)) \end{aligned}$$

Damit kann man nun eine renormierte Hamiltonfunktion schreiben als:

$$\tilde{\mathcal{H}} = A + \tilde{K} \sum_{nn(i)} \tilde{s}_i \tilde{s}_j + \tilde{L} \sum_{nnn(i)} \tilde{s}_i \tilde{s}_j + C \sum_{4} \tilde{s}_i \tilde{s}_j \tilde{s}_l \tilde{s}_m$$

Der letzte Term wird vernachlässigt (“Trunkation”). Der vorletzte Term beschreibt Wechselwirkungen mit den übernächsten Nachbarn. In der ursprünglichen Hamiltonfunktion waren derartige Wechselwirkungen noch nicht enthalten.
Schneidet man bei jeder Renormierung die Terme mit den höheren Wechselwirkungen ab, dann enthält jede Hamiltonfunkton nur die Wechselwirkungen mit den nächsten und den übernächsten Nachbarn. Man kann die Renormierung der Hamiltonfunktion dann betrachten als eine Änderung der Wechselwirkungskonstanten $K \to \tilde{K}$ und $L \to \tilde{L}$ unter der Skalentransformation $x \to lx$.

$$\begin{aligned} \tilde{K} &= \frac{1}{4} \ln\cosh(4K) + L \\ \tilde{L} &= \frac{1}{8} \ln\cosh(4K) \end{aligned}$$

Man nennt das obige Vorgehen zur Ermittlung der renormierten Hamiltonfunktion eine *Blockspinrenormierung*.
Führt man diese Transformation jetzt n-mal hintereinander aus, und verfolgt den “Weg” der Punkte (K,L) in der K-L-Ebene für viele Anfangsbedingungen, so bekommt man das in der Abbildung gezeigte Flußdiagramm.

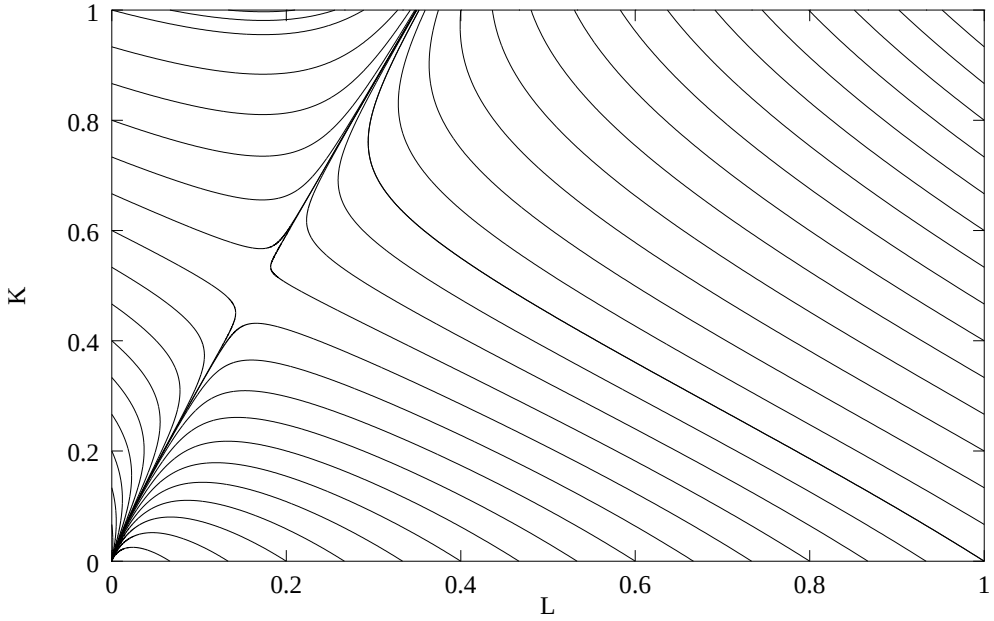


Abb. 7.6 *Flußdiagramm der Transformation*

Die Transformation besitzt Fixpunkte für $K, L = 0, \pm\infty$ und

$$K^* = \frac{3}{8} \ln\cosh(4K^*)$$

$$L^* = \frac{1}{3}K^*$$

Durch den Fixpunkt geht die *kritische Fläche.* Alle Transformationen mit einem Startpunkt auf dieser Fläche laufen in den Fixpunkt hinein. Alle Transformationen mit einem Startpunkt außerhalb bewegen sich vom Fixpunkt weg. Die Richtung senkrecht zur kritischen Fläche im Fixpunkt bezeichnet man als *relevante* Richtung. Parallel zur kritischen Fläche spricht man von einer *irrelevanten* Richtung.
Um ein Skalengesetz der Form

$$\tilde{K} - K^* = l^{y_T}(K - K^*)$$

zu erhalten, macht man nun eine Taylorentwicklung der obigen Transformationsformeln bis zu den linearen Termen. Mit den Abkürzungen $\delta K = K - K^*$ und $\delta L = L - L^*$ kann man dann die Transformation in einer Matrixschreibweise darstellen.

$$\begin{pmatrix} \delta K_{n+1} \\ \delta L_{n+1} \end{pmatrix} = \begin{pmatrix} 4K^* & 1 \\ 2K^* & 0 \end{pmatrix} \begin{pmatrix} \delta K_n \\ \delta L_n \end{pmatrix}.$$

Die Transformationsmatrix, die Jacobimatrix, soll als $T_{\alpha\beta}$ abgekürzt werden. Die Eigenvektoren von $T_{\alpha\beta}$ zeigen gerade in die relevante bzw. irrelevante Richtung.
In Eigenvektorrichtung gilt

$$\begin{pmatrix} \delta K_{n+1} \\ \delta L_{n+1} \end{pmatrix} = \lambda \begin{pmatrix} \delta K_n \\ \delta L_n \end{pmatrix}.$$

In der relevanten Richtung, in der sich die Transformation vom kritischen Punkt wegbewegt, muß für den Eigenwert λ somit gelten:

$$|\lambda_r| > 1$$

In der irrelevanten Richtung ist der Betrag des Eigenwertes entsprechend kleiner 1. Interessant sind jedoch nur die relevanten Richtungen. Der Eigenwert in der relevanten Richtung liefert den kritischen Exponenten.

$$\delta K_{n+1} = \lambda_r \delta K_n = l^{y_T} \delta K_n \qquad \text{mit } y_T = \frac{1}{\nu}$$

Allgemeinere Formulierung

Zum Abschluß soll das vorgestellte Verfahren noch einmal allgemeiner formuliert werden. Man schreibt die Hamiltonfunktion in der Form:

$$\mathcal{H} = \sum_{\alpha=1}^{M} K_\alpha O_\alpha \qquad \text{mit } O_\alpha = \prod_{i \in \alpha} s_i$$

O_2 ist die Wechselwirkung mit nächsten Nachbarn, O_3 mit übernächsten Nachbarn, usw. Die Renormierungsgleichung hat die Form:

$$\exp(G+\tilde{\mathcal{H}}(\tilde{s})) = \sum_{\{s\}} P(\tilde{s},s)\exp(\mathcal{H}(s))$$

G ist ein zusätzlicher, konstanter Parameter der Transformation. In obigem Beispiel galt $G=0$. P ist eine Gewichtsfunktion mit $P(\tilde{s},s)>0$. Im vorigen Beispiel war $P(\tilde{s},s)=1$ falls $\tilde{s}$ und s zu verschiedenen Untergittern gehörten und Null sonst. Die Zustandssumme des renormierten Systems soll gleich der Zustandssumme des ursprünglichen Systems sein,

$$\begin{aligned} & & \tilde{Z} &= Z \\ \Leftrightarrow \quad & \sum_{\{\tilde{s}\}} \exp(G+\tilde{\mathcal{H}}(\tilde{s})) &= \sum_{\{s\}} \exp(\mathcal{H}(s)) \\ & \text{mit} \quad \sum_{\{\tilde{s}\}} P(\tilde{s},s) &= 1 \end{aligned}$$

Diese Bedingung ist gleichbedeutend zur Erhaltung der freien Energie.
Man nimmt nun an, $\tilde{\mathcal{H}}$ habe die gleiche Form wie $\mathcal{H}$, d.h.

$$\tilde{\mathcal{H}} = \sum_{\alpha=1}^{M} \tilde{K}_\alpha \tilde{O}_\alpha$$

Beide Hamiltonfunktionen sollen nur M Terme besitzen. Die Beiträge aller weiteren Terme werden vernachlässigt. Einsetzen in die Renormierungsgleichung liefert die Transformationsgleichungen der K_α:

$$\tilde{K}_\alpha = \tilde{K}_\alpha(K_\alpha)$$

Die Jacobimatrix der Transformationsgleichungen am Fixpunkt

$$T_{\alpha\beta} = \frac{\partial \tilde{K}_\alpha}{\partial K_\beta}$$

charakterisiert eindeutig das Systemverhalten in der Nähe des kritischen Punktes K^*_α,

$$\tilde{K}_\alpha - K^*_\alpha = \sum_\beta T_{\alpha\beta}|_{K^*}(K_\beta - K^*_\beta).$$

Die Matrix T läßt sich diagonalisieren und aus den Eigenwerten ergeben sich dann die kritischen Exponenten.

7.4 Monte Carlo Renormierungsgruppen

Die Theorie der Renormierungsgruppen stammt von Ma (1977) und Swendsen (1979). Man bestimmt die Mittelwerte der Wechselwirkungsoperatoren O_α .

$$\langle O_\alpha \rangle = \frac{\sum_{\{s\}} O_\alpha \exp(\sum_\beta K_\beta O_\beta)}{\sum_{\{s\}} \exp(\sum_\beta K_\beta O_\beta)}$$

$\sum_{\{s\}}$ ist die Summe über alle Konfigurationen. Es gilt:

$$\langle O_\alpha \rangle = \frac{\partial F}{\partial K_\alpha}$$

$F = \ln Z$ ist die freie Energie. Des weiteren gelten die Beziehungen:

$$\begin{aligned} \frac{\partial \langle O_\alpha \rangle}{\partial K_\beta} &= \langle O_\alpha O_\beta \rangle - \langle O_\alpha \rangle \langle O_\beta \rangle = \chi_{\alpha\beta} \\ \frac{\partial \langle \tilde{O}_\alpha \rangle}{\partial K_\beta} &= \langle \tilde{O}_\alpha O_\beta \rangle - \langle \tilde{O}_\alpha \rangle \langle O_\beta \rangle = \tilde{\chi}_{\alpha\beta} \end{aligned}$$

Die partiellen Ableitungen nach K entsprechen somit den Korrelationsfunktionen zwischen verschiedenen Operatoren. Die Korrelationsfunktionen sind über $O_\alpha = \prod_{i \in \alpha} s_i$ numerisch bestimmbar. Damit erhält man mit

$$\frac{\partial \langle \tilde{O}_\alpha^{(n)} \rangle}{\partial K_\beta} = \sum_\gamma \frac{\partial \tilde{K}_\gamma}{\partial K_\beta} \frac{\partial \langle \tilde{O}_\alpha^{(n)} \rangle}{\partial \tilde{K}_\gamma} = \sum_\gamma T_{\gamma\beta} \chi_{\alpha\gamma}^{(n)}$$

auch die Jacobimatrix und daraus, wie oben beschrieben, die Eigenwerte und kritischen Exponenten.

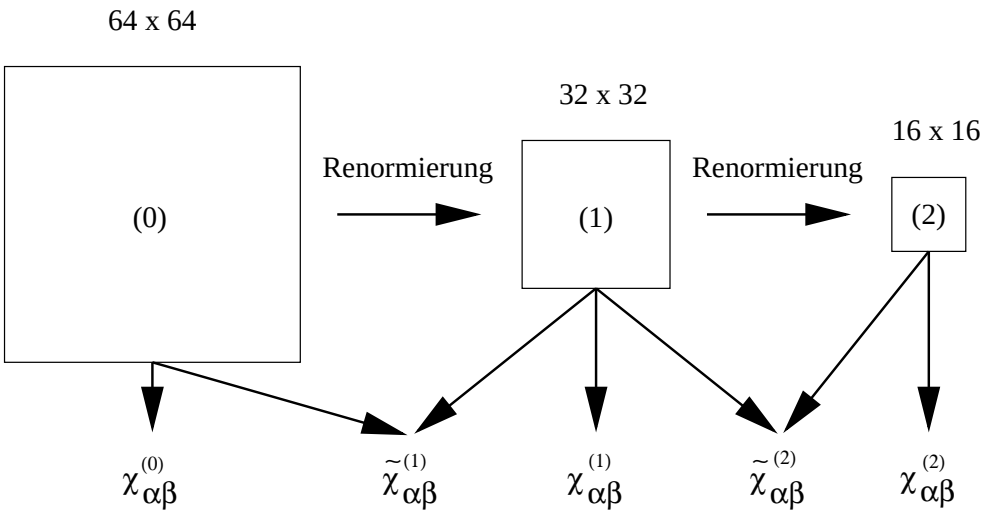


Abb. 7.7 *Bestimmung der Korrelationsfunktionen bzw. der partiellen Ableitungen aus dem renormierten System*

Man startet mit einem System bei der bekannten, kritischen Temperatur T_c. Durch wiederholte Renormierung bewegt man sich auf der kritischen Ebene auf den kritischen Punkt zu. Da man i.A. nicht exakt auf der kritischen Ebene starten wird, bewegt man sich nur für eine begrenzte Anzahl von Renormierungen auf den Fixpunkt zu, danach entfernt man sich wieder von ihm. Aus den Systemen kann man nun nach obigen Formeln die Jacobimatrix und die kritischen Exponenten bestimmen (siehe Abbildung). Die besten Werte erhält man im Punkt größter Annäherung.
Die Fehlerquellen dieses Verfahrens sind:

1. Statistik
2. Trunkation
3. Anzahl der Iterationen, Endlichkeit des Systems, Ungenauigkeit von T_c, welches sich bei der analytischen Methode eigentlich gerade aus der Analyse, wenn auch nur näherungsweise, *ergibt.*

Kapitel 8

Molekulardynamik

8.1 Eigenschaften der Molekulardynamik

Der Begriff Molekulardynamik (Molekulardynamik) hat sich in der Physik zur Bezeichung numerische Lösungsverfahren zur expliziten Berechnung der Bewegung eines Systems von vielen Teilchen eingebürgert. Als Beispiele für solche Systeme seien angeführt:

- Das Edelgas Argon. Die Edelgasatome sind im Grundzustand neutral und haben annähernd Kugelgestalt. Aus Simulationen mit ca. 30 Atomen können bereits Aussagen über thermodynamische Zustandsgrößen gewonnen werden.
- Polymerketten
- Moleküle wie z.B. Wasser, aber auch komplexe Proteine oder Moleküle (drug design)
- Granulare Materialien wie z.B. Sand. Die Molekulardynamik liefert hier Aussagen zur Strukturbildung.

Im Gegensatz zu den in den vorigen Kapiteln behandelten Simulationsmethoden ist die Molekulardynamik rein deterministisch (bis auf die numerische Ungenauigkeit). Zufallszahlen finden nur noch bei der Generierung des Anfangszustandes Anwendung.

8.2 Bewegungsgleichung eines N-Teilchensystems

Ein einzelnes Teilchen der Molekulardynamik bewegt sich gemäß den Gesetzen der Newton'schen Mechanik:

- Ein Teilchen, auf das keine Kräfte wirken, bewegt sich geradlinig und gleichförmig.
- Es gilt actio = reactio.

- Mehrere auf ein Teilchen einwirkende Kräfte addieren sich vektoriell, und das Teilchen bewegt sich dann gemäß $\vec{F} = m\vec{a}$ bzw. der Bewegungsgleichungen, die sich aus der Lagrange- oder Hamiltonformulierung ergeben (s.u.).

Ist das Gesamtsystem abgeschlossen, an ein Wärmebad angeschlossen oder an ein Wärme- und ein Teilchenbad angeschlossen, gelten außerdem die Zwangsbedingungen der Statistischen Mechanik für mikrokanonische (konstante Energie und Teilchenzahl), kanonische (konstante Temperatur und Teilchenzahl) bzw. großkanonische Systeme (konstante Temperatur, konstantes chemisches Potential). Hierbei muß man die Einschränkung machen, daß diese Zwangsbedingungen in Exaktheit nur für sehr viele Teilchen ($\approx 10^{23}$) gelten, während man sich bei numerischen Simulationen auf einige 10^3 Teilchen beschränken muß.
Das Teilchen i sei durch die generalisierten Koordinaten und Impulse

$$\vec{q}_i = (q_i^1, \ldots, q_i^\alpha) \quad , \qquad \vec{p}_i = (p_i^1, \ldots, p_i^\alpha)$$

beschrieben. Die Koordinaten und Impulse des Gesamtsystems von N Teilchen sind dann durch die Vektorfelder

$$Q = (\vec{q}_1, \ldots, \vec{q}_N) \quad , \qquad P = (\vec{p}_1, \ldots, \vec{p}_N)$$

gegeben. Die Hamiltonfunktion ist meist von der Form

$$\mathcal{H}(P,Q) = \mathcal{K}(P) + \mathcal{V}(Q)$$

$\mathcal{K}$ ist dabei die kinetische Energie mit

$$\mathcal{K}(P) = \sum_i \sum_{k=1}^{\alpha} \frac{p_i^{k\,2}}{2m_i}$$

wobei m_i die Masse des Teilchens i ist. Die potentielle Energie läßt sich durch eine Taylorentwicklung in 1-, 2- ,3- , ... Teilchenwechselwirkungsterme zerlegen:

$$\mathcal{V} = \sum_i v_i(q_i) = \sum_i \sum_{j>i} v_2(q_i, q_j) + \sum_i \sum_{j>i} \sum_{k>j} v_3(q_i, q_j, q_k) + \ldots$$

Höhere Wechselwirkungen als die 3-Teilchenwechselwirkung werden in der Regel vernachlässigt. Die 2-, 3-, ... Teilchenwechselwirkungen faßt man meistens in einen effektiven Zweiteilchenwechselwirkungsterm $v_2^{eff}(q_i, q_j)$ zusammen, welcher oft nur vom Abstand $r = |q_i - q_j|$ abhängt und in einen anziehenden (attraktiven) und einen abstoßenden (repulsiven) Beitrag zerlegt werden kann:

$$v_2^{eff}(q_i, q_j) = v^{att}(r) + v^{rep}(r)$$

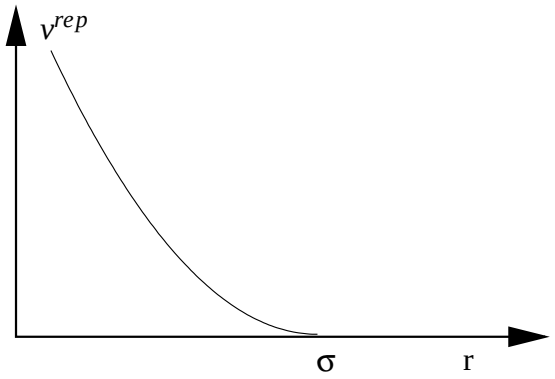
8.2.1 Abstoßende Potentiale

1. Das *hard-core* Potential

$$v^{rep}(r) = \begin{cases} \infty & r < \sigma \\ 0 & r \geq \sigma \end{cases}$$

Das hard-core Potential beschreibt harte Kugeln. Für Atome ist typischerweise $\sigma \approx 0.35nm$.

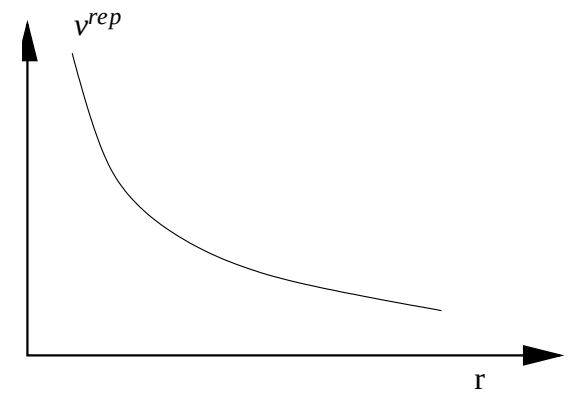
2. Das elastische Potential



$$v^{rep}(r) = \begin{cases} \frac{k}{2}(\sigma - r)^2 & r < \sigma \\ 0 & r \geq \sigma \end{cases}$$

Das elastische Potential ist der abstoßende Teil des Potentials einer Hookschen Feder mit der Federkonstanten k. Man modelliert mit diesem Potential abstoßende Kräfte, die linear mit dem Überlapp zweier Teilchen anwachsen ($\sigma = R_1 + R_2$ mit den Teilchenradien R_1 und R_2).

3. Das *soft-core* Potential



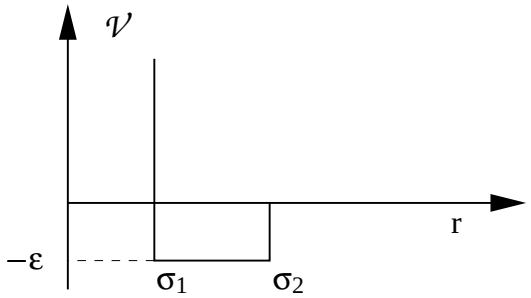
$$v^{rep}(r) = \varepsilon(\frac{\sigma}{r})^{\nu}$$

Der Abstoßungsparameter ν ist typischerweise 12 (steiles Potential).

$\nu = 1$ ist der Fall elektrostatischer (langreichweitiger) Abstoßung. Langsam abklingende Potentiale wie das Gravitations-/elektrostatische Potential werfen in numerischen Simulationen einige Probleme auf. Ist das simulierte System klein, hat die Wechselwirkung im Abstand $r \approx L$ (L = Systemabmessung) immer noch einen signifikanten Wert. Bei periodisch fortgesetzten Systemen kann das zu einer Wechselwirkung des Teilchens mit sich selbst führen.

8.2.2 Potentiale mit attraktivem Anteil

Diese Potentiale beschreiben die chemische Bindung (kovalent, ionisch), sowie van der Waalskräfte und im makroskopischen Fall Kapillarkräfte. Für viele Anwendungen zu einfach sind Kastenpotentiale:

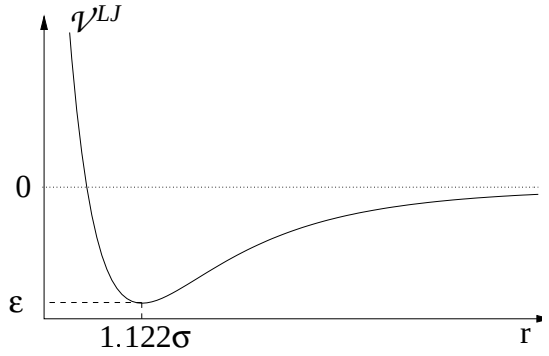


$$\mathcal{V}(r) = \begin{cases} \infty & r < \sigma_1 \\ -\varepsilon & \sigma_1 \le r < \sigma_2 \\ 0 & r \ge \sigma_2 \end{cases}$$

Dieses Potential hat bei der numerischen Behandlung den Vorteil, daß sich das Teilchen an allen Orten $r \neq \sigma_1, \sigma_2$ gleichförmig und geradlinig bewegt, man in der Simulation also keine Kräfte berechnen muß.

Das einfachste realistische Potential, welches sehr gut das Argon beschreibt (van der Waalskräfte), ist das Lenard-Jones Potential

$$\mathcal{V}^{LJ}(r) = 4\varepsilon\left[(\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^{6}\right]$$



8.2.3 Bewegungsgleichungen

Die Hamiltonschen Bewegungsgleichungen lauten:

$$\dot{q}_k = \frac{\partial \mathcal{H}}{\partial p_k} \qquad \dot{p}_k = -\frac{\partial \mathcal{H}}{\partial q_k}$$

Nimmt man z.B. als Koordinaten den Ortsvektor $\vec{q}_i = \vec{r}_i$, also $\dot{\vec{q}}_i = \vec{v}_i$, so ergeben sich zwei Gleichungen 1. Ordnung

$$\dot{\vec{r}}_k = \vec{v}_k = \frac{\vec{p}_k}{m_k} \qquad \dot{\vec{p}}_k = -\frac{\partial \mathcal{V}(q_k)}{\partial q_k} = \vec{f}_k$$

wobei $\vec{f}_k$ die Kraft ist. Dies kann man in der Newtonschen Bewegungsgleichung (2. Ordnung) zusammenfassen

$$m_i \dot{\vec{v}}_i = \vec{f}_i$$

Die Wechselwirkung mit Wänden beschreibt man ebenfalls durch Potentiale (meist die gleichen, wie die der Teilchen), wobei r dann der euklidische Abstand zur Wand ist.

8.2.4 Erhaltungssätze

- Energieerhaltung gilt immer (solange $\mathcal{K}$ und $\mathcal{V}$ nicht explizit von der Zeit abhängen).
- Der Gesamtimpuls

  $$\vec{P} = \sum_i \vec{p}_i$$

  ist erhalten, solange das System keine Wände hat, also z.B. bei periodischen Randbedingungen.

- Der Gesammtdrehimpuls

$$\vec{L} = \sum_i \vec{r}_i \times \vec{p}_i$$

  um den Mittelpunkt des Systems ist nur in kugelförmigen Behältern erhalten (exotisch).

- Es gilt die Zeitumkehrinvarianz.

8.2.5 Kontaktzeit

Über die Energie und das Wechselwirkungspotential kann man auch die Kontaktzeit t_c zwischen zwei Teilchen berechnen. Die Berechnung erfolgt der Einfachheit halber eindimensional. Es soll Energieerhaltung gelten, d.h.

$$E = \frac{1}{2} m\dot{r}^2 + \mathcal{V}(r) = const.$$

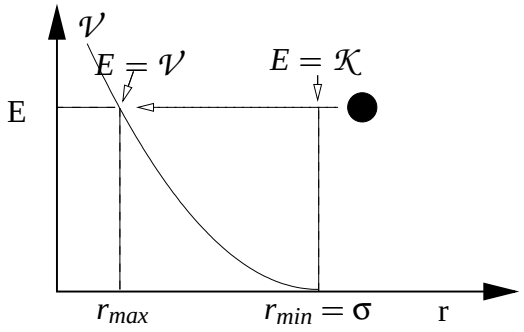
Aufgelöst nach der Geschwindigkeit ergibt sich daraus:

$$\frac{dr}{dt} = \left[\frac{2}{m}(E - \mathcal{V}(r))\right]^{\frac{1}{2}}$$

Die Lösung ergibt sich durch Trennung der Variablen:

$$\frac{1}{2}t_c = \int_0^{\frac{1}{2}t_c} dt = \int_{r_{min}}^{r^{max}} \left[\frac{2}{m}(E - \mathcal{V}(r))\right]^{-\frac{1}{2}} dr$$

Als Untergrenze r_{min} des Integrals über den Abstand wählt man bei einem "abgschnittenen" Potential, wie z.B. dem elastischen Potential, den Anfangspunkt σ. Bei einem Potential mit attraktivem Anteil, wie z.B. dem Lennard-Jones-Potential, beginnt man die Integration im Minimum des Potentials. Die Obergrenze r_{max} ist durch den Umkehrpunkt des Teilchens gegeben. Diesen erhält man aus der Bedingung $\mathcal{V}(r_{max}) = E = \frac{v^2}{2m}$.



8.3 Integration der Bewegungsgleichungen

8.3.1 Allgemeines

Die Grundidee zur numerischen Lösung der Bewegungsgleichungen eines Systems aus N Teilchen besteht darin, zunächst die Zeit zu diskretisieren, indem man nur noch Zeitpunkte in Abständen Δt betrachtet. Die Geschwindigkeitsänderung eines Teilchens i am Ort $\vec{x}_i$ berechnet sich damit zu

$$\Delta\vec{v}_i = \frac{\vec{f}_i(\vec{x}_i(t))}{m_i}\Delta t \qquad .$$

Den neuen Ort kann man dann mit

$$\vec{x}_i(t+\Delta t) = \vec{x}_i(t) + \Delta t\vec{v}_i(t)$$

berechnen. Die Ergebnisse dieses naiven Zugangs, der sogenannten *Euler*-Methode sind recht schlecht mit einem Fehler $\propto O(\Delta t^2)$. Es gibt jedoch noch viele andere finite Differenzen Algorithmen. Kriterien für einen guten Algorithmus sind:

- Ein einfacher Iterationsschritt und wenig Speicherbedarf
- Δt sollte so groß wie möglich sein, damit man lange Systemzeiten in möglichst kurzer Rechenzeit simulieren kann. Im Allgemeinen muß man Δt kleiner machen, wenn das Potential steiler ist, oder die Geschwindigkeiten größer sind, d.h. für höhere Temperaturen oder leichtere Teilchen. In jedem Fall zu steil sind jedoch Hard-core und Kastenpotentiale. Sie können mit der Methode der finiten Differenzen nicht behandelt werden, da sie unendliche Kräfte am Teilchenrand produzieren.
- Die Trajektorien der Teilchen sollten so gut wie möglich reproduzierbar sein.
- Die oben formulierten Erhaltungsätze müssen erfüllt sein.
- Das Programm sollte einfach sein und gegebenenfalls parallelisiert werden können.

Die Größe von Δt kann man durch folgende Überlegung abschätzen: Zwei auf Kollisionskurs fliegende Teilchen dürfen sich nicht durchdringen, sondern müssen einen Stoß ausführen. Damit dieser Stoß noch ausreichend aufgelöst wird bzw. damit die Teilchen sich überhaupt spüren, dürfen sie pro Zeitschritt Δt nicht mehr als ca. ein Zehntel ihres Durchmessers d zurücklegen. Damit erhält man für Δt die Abschätzung

$$\Delta t = \frac{1}{10}\frac{d}{v_{max}}.$$

v_{max} ist die maximale Geschwindigkeit, die ein Teilchen besitzen kann. Die Tabelle gibt die Größenordnung von Δt für verschiedene Systeme wieder. Eingetragen sind auch die relevanten Zeiträume und die Anzahl der Iterationsschritte, die nötig wären, diese zu simulieren. Man sieht, daß man die gewünschten Zeiten nur bei den granularen Materialien in endlicher Zeit simulieren kann.

	d [m]	v_{max} [$\frac{m}{s}$]	Δt [s]	Simulationszeit [s]	Iterationen
Atome und Moleküle	10^{-10}	10	10^{-12}	1	10^{12}
granulare Materialien	10^{-3}	1	10^{-4}	10^{2}	10^{6}
Astrophysik	10^{7}	10^{8}	10^{-2}	10^{13}	10^{15}

Außer in wenigen Fällen ($N = 2$ oder Kette harmonischer Oszillatoren) sind Vielteilchensysteme chaotisch: Man definiert den Abstand zwischen zwei Konfigurationen $\vec{r}_i^{\,1}(t)$ und $\vec{r}_i^{\,2}(t)$ durch

$$(\Delta r(t))^2 = \frac{1}{N}\sum_i \left(\vec{r}_i^{\,1}(t) - \vec{r}_i^{\,2}(t)\right)^2$$

Im chaotischen Fall divergiert dieses $\Delta r(t)$ stets exponentiell mit der Zeit unabhängig davon, wie klein der Anfangsunterschied $\Delta r(0)$ war. Da jeder Rechner Rundungsfehler hat, ist es unmöglich über länger als typischerweise mehrere hundert Iterationen die Dynamik des Systems zu reproduzieren. Das ist nicht so schlimm, da man sowieso meist nur an statistischen Ensembles interessiert ist, so daß der wesentliche Punkt darin besteht, daß die makroskopischen Größen, wie z.B. die Energie, erhalten sind.
Viele gute Algorithmen sind bekannt, einige um das System 1. Ordnung zu lösen, andere um die Newtonschen Gleichungen (2. Ordnung) direkt zu lösen. Im Folgenden werden verschiedene Algorithmen vorgestellt und miteinander verglichen.

8.3.2 Das Runge-Kutta Verfahren

Das Runge-Kutta Verfahren dient zur Berechnung von Differentialgleichungen der Form

$$\dot{y}_i(t) = g(t, y_1 \ldots y_N)$$

Die oben vorgestellte Euler-Methode löste diese Gleichung, indem die Ableitung zur Zeit t berechnet wurde und dann ausgehend vom Punkt $y(t)$ um $\Delta t g(t)$ "weitergegangen" wurde. Das Runge-Kutta Verfahren 2. Ordnung führt nun zuerst einen Versuchsschritt in die Mitte des Intervals Δt aus, bestimmt dort den Wert von g und führt erst damit den tatsächlichen Schritt aus.

$$\begin{aligned} y_i(t+\frac{1}{2}\Delta t) &= y_i(t) + \frac{1}{2}\Delta t g(t, y_j(t)) \\ y_i(t+\Delta t) &= y_i(t) + \Delta t g(t+\frac{1}{2}\Delta t, y_j(t+\frac{1}{2}\Delta t)) \end{aligned}$$

j nimmt dabei die Werte von $1 \ldots N$ an. Angewandt auf das hier behandelte N-Teilchen System kann man somit schreiben:

$$\begin{aligned} \vec{v}_1 &= \vec{v}(t) + \frac{1}{2}\Delta t \frac{\vec{f}(t)}{m} \\ \vec{x}(t+\Delta t) &= \vec{x}(t) + \Delta t \vec{v}_1 \end{aligned}$$

Der Index *i* des Teilchens ist hier weggelassen. Für eine weitere Verbesserung des Verfahrens kann man noch weitere Hilfspunkte innerhalb Δt heranziehen. Am häufigsten verwendet man ein Runge-Kutta Verfahren 4.Ordnung.

$$\begin{aligned}
\vec{v}_1 &= \vec{v}(t) + \Delta t \frac{\vec{f}(\vec{x}(t))}{m} \\
\vec{x}_1 &= \Delta t \vec{v}(t) \\
\vec{v}_2 &= \vec{v}(t) + \Delta t \frac{\vec{f}(\vec{x}(t) + \frac{\vec{x}_1}{2})}{m} \\
\vec{x}_2 &= \Delta t \vec{v}_1 \\
\vec{v}_3 &= \vec{v}(t) + \Delta t \frac{\vec{f}(\vec{x}(t) + \frac{\vec{x}_2}{2})}{m} \\
\vec{x}_3 &= \Delta t \vec{v}_2 \\
\vec{x}_4 &= \Delta t \vec{v}_3
\end{aligned}$$

Dann berechnet man die neue Position durch

$$\vec{x}(t+\Delta t) = \vec{x}(t) + \frac{\vec{x}_1}{6} + \frac{\vec{x}_2}{3} + \frac{\vec{x}_3}{3} + \frac{\vec{x}_4}{6} + O(\Delta t^5) \quad .$$

An dieser Stelle sei jedoch angemerkt, daß eine Erhöhung der Ordnung nicht unbedingt zu einer Verbesserung des Ergebnisses führen muß. Für eine detailliertere Beschreibung des Verfahrens siehe [14].

8.3.3 Die Verlet Methode (1967)

Man entwickelt $\vec{x}(t \pm \Delta t)$ nach Taylor

$$\begin{aligned}
\vec{x}(t+\Delta t) &= \vec{x}(t) + \Delta t \vec{v} + \frac{1}{2}\Delta t^2 \vec{a}(t) \\
\vec{x}(t-\Delta t) &= \vec{x}(t) - \Delta t \vec{v} + \frac{1}{2}\Delta t^2 \vec{a}(t)
\end{aligned}$$

und addiert beide Gleichungen

$$\vec{x}(t+\Delta t) = 2\vec{x}(t) - \vec{x}(t-\Delta t) + \Delta t^2 \vec{a}(t)$$

Dies ergibt die Verlet'sche Iterationsmethode, welche einen Fehler der Ordnung $O(\Delta t^4)$ hat. Die Geschwindigkeiten wurden eliminiert, können jedoch aus

$$\vec{v}(t) = \frac{\vec{x}(t+\Delta t) - \vec{x}(t-\Delta t)}{2\Delta t}$$

berechnet werden. Dieser Algorithmus für Gleichungen zweiter Ordnung erfüllt automatisch die Zeitumkehrinvarianz. Er hat daher die häufig wünschenswerte Eigenschaft,

dass sich die Effekte kleiner numerischer Fehler aufgrund endlicher Zeitschrittweite nicht systematisch aufsummieren: z.B. driftet die Energie nicht in eine Richtung, sondern führt eine Art random walk kleiner Schrittweite um einen Mittelwert herum aus.
Dieser Algorithmus erfordert die Speicherung von $3\alpha N$ Variablen, da man drei verschiedene Zeitpunkte betrachtet. Man kann den Algorithmus auch systematisch verbessern, indem man noch mehr Zeitpunkte hinzunimmt, doch der höhere Rechenaufwand lohnt sich meist nicht.
Ein Problem dieser Methode ist, daß man große Zahlen $O(\Delta t^0)$ und $O(\Delta t^2)$ miteinander addiert, was numerisch zu größeren Rundungsfehlern führt.

8.3.4 Die leap-frog Methode (1970)

Die "leap-frog" Methode von Hockney (1970) ist eine Abwandlung des Verlet Algorithmus. Sie umgeht das Problem der Addition von Zahlen verschiedener Größenordnung. Die Geschwindigkeiten werden jetzt zu halben Zeitschritten betrachtet:

$$\vec{v}(t+\frac{1}{2}\Delta t) = \vec{v}(t-\frac{1}{2}\Delta t) + \Delta t\vec{a}(t)$$

Der neue Ort des Teilchens ergibt sich nun aus

$$\vec{x}(t+\Delta t) = \vec{x}(t) + \Delta t\vec{v}(t+\frac{1}{2}\Delta t)$$

Diese Methode behandelt zwei Gleichungen 1. Ordnung und berechnet die Geschwindigkeiten explizit, was für Simulationen bei konstanter Temperatur von Vorteil ist.
In der Praxis ist der Unterschied zwischen leap-frog und der Methode von Euler geringer, als es hier den Anschein hat. Die Euler Methode führt die Berechnung der neuen Werte in der Reihenfolge

$$\begin{aligned}
\vec{a}(t+\Delta t) &= \frac{\vec{f}(\vec{x}(t))}{m} \\
\vec{x}(t+\Delta t) &= \vec{x}(t) + \Delta t\ \vec{v}(t) \\
\vec{v}(t+\Delta t) &= \vec{v}(t) + \Delta t\ \vec{a}(t+\Delta t)
\end{aligned}$$

aus, während man die leap-frog Methode in der Form

$$\begin{aligned}
\vec{a}(t+\Delta t) &= \frac{\vec{f}(\vec{x}(t))}{m} \\
\vec{v}(t+\frac{1}{2}\Delta t) &= \vec{v}(t-\frac{1}{2}\Delta t) + \Delta t\ \vec{a}(t+\Delta t) \\
\vec{x}(t+\Delta t) &= \vec{x}(t) + \Delta t\ \vec{v}(t+\frac{1}{2}\Delta t)
\end{aligned}$$

implementiert. Trotz dieses geringen Unterschieds liefert diese Methode mit ihrer "natürlichen" Reihenfolge der Berechnung die besseren Ergebnisse.

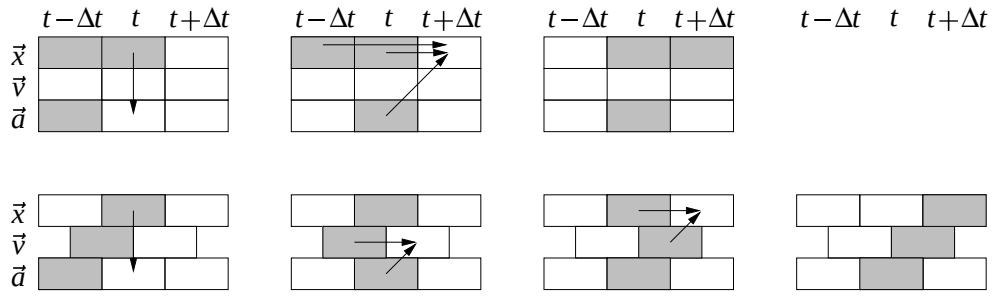


Abb. 8.1 *Veranschaulichung der Vorgehensweise der Verlet-Methode (obere Reihe) und der leap-frog Methode (untere Reihe) (nach [15])*

8.3.5 Prediktor-Korrektor Methode

Bei der Prediktor-Korrektor Methode betrachtet man mehrere zeitliche Ableitungen und macht zunächst eine Voraussage für den neuen Wert durch eine Taylorentwicklung:

$$\begin{aligned}
\vec{x}^P(t+\Delta t) &= \vec{x}(t)+\Delta t\vec{v}(t)+\frac{\Delta t^2}{2}\vec{a}(t)+\frac{\Delta t^3}{6}\ddot{\vec{v}}(t)+O(\Delta t^4)\\
\vec{v}^P(t+\Delta t) &= \vec{v}(t)+\Delta t a(t)+\frac{\Delta t^2}{2}\ddot{\vec{v}}(t)\\
\vec{a}^P(t+\Delta t) &= \vec{a}(t)+\Delta t\ddot{\vec{v}}(t)\\
\ddot{\vec{v}}^P(t+\Delta t) &= \ddot{\vec{v}}(t),
\end{aligned}$$

wobei dies ein Prediktor-Korrektor der 4. Ordnung ist, da der Fehler $O(\Delta t^4)$ ist. Allerdings werden die Trajektorien durch die in der Zwischenzeit wirkenden Kräfte verformt. Man berechnet die korrekte Beschleunigung aus dem Newtonschen Gesetz:

$$\dot{\vec{v}}_i^C(t+\Delta t)=\frac{F_i(t+\Delta t)}{m_i}=g+\frac{1}{m_i}\sum_j f_{ij}(\vec{x}_i^P(t+\Delta t)-\vec{x}_j^P(t+\Delta t))$$

Dabei werden die vorhergesagten Werte zur Zeit $t+\Delta t$ benutzt. Man definiert die Korrektur durch

$$\delta_i(t+\Delta t)=\dot{\vec{v}}_i^C(t+\Delta t)-\dot{\vec{v}}_i^P(t+\Delta t)$$

und bestimmt die korrigierten Werte in niedrigster Ordnung durch

$$\begin{aligned}
\vec{x}_i^C(t+\Delta t) &= \vec{x}_i^P(t+\Delta t)+c_0\delta_i(t+\Delta t)\\
\vec{v}_i^C(t+\Delta t) &= \vec{v}_i^P(t+\Delta t)+c_1\delta_i(t+\Delta t)\\
\vec{a}_i^C(t+\Delta t) &= \vec{a}_i^P(t+\Delta t)+c_2\delta_i(t+\Delta t)\\
\ddot{\vec{v}}_i^C(t+\Delta t) &= \ddot{\vec{v}}_i^P(t+\Delta t)+c_3\delta_i(t+\Delta t)
\end{aligned}$$

Gear (1971) hat die Konstanten $c_0 = 1/6$, $c_1 = 5/6$, $c_2 = 1$ und $c_3 = 1/3$ aus einem Variationsprinzip bestimmt. Für Prediktor-Korrektor anderer Ordnung oder für Differentialgleichungen erster Ordnung findet man andere Konstanten, welche z.B. in [15] tabelliert sind. Es existieren auch andere Prediktor-Korrektor Schemata, wie die von Beeman und Toxvaerd.

Es gibt numerische Integrationsverfahren, deren mathematische Eigenschaften besser sind als die von Prediktor-Korrektor Integratoren. Diese haben aber üblicherweise den Nachteil, pro Zeitschritt mehr als eine oder gar eine iterative Auswertung der Kräfte zu erfordern. Da diese Berechung den Großteil der Rechenzeit eines MD Algorithmus umfaßt, ergeben sich meist Vorteile für das auch konzeptionell einfache Prediktor-Korrektor Verfahren.

8.3.6 Fehlerabschätzung

Die Präzision einer MD Rechnung kann man durch die Energiefluktuation

$$(\delta E)^2 = \langle E^2 \rangle_T - \langle E \rangle_T^2$$

über ein festes Zeitinterwall T messen.

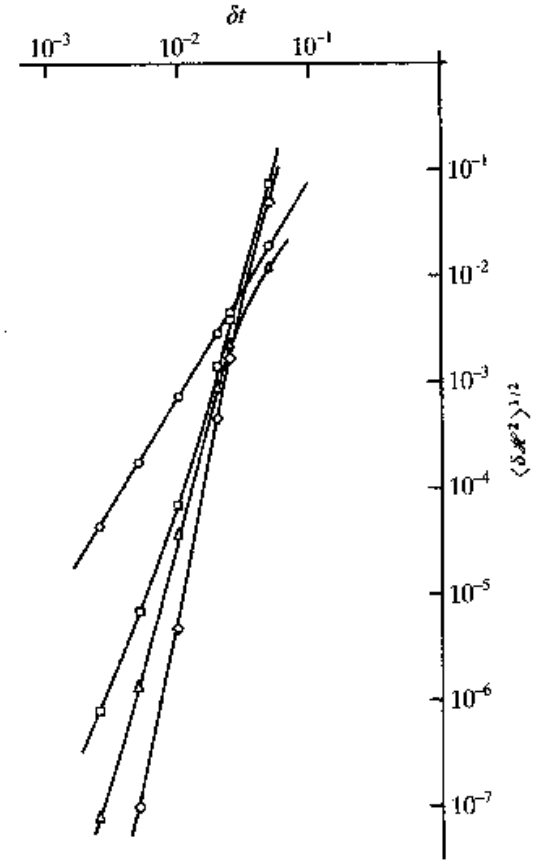


Abb. 8.2 *Vergleich der Energieerhaltung verschiedener Finite-Differenzen Verfahren. Die Kurven entsprechen Verlet (Kreise), Prediktor-Korrektor vierter Ordnung (Quadrate), Prediktor-Korrektor fünfter Ordnung (Dreiecke) und Prediktor-Korrektor sechster Ordnung (Rauten) (nach [15]).*

Die Abbildung vergleicht Verlet und Prediktor-Korrektor bei gleicher Simulationsdauer T. Für festes Δt muß man $n = T/\Delta t$ Iterationen machen, so daß Δt der benötigten CPU-Zeit umgekehrt proportional ist. Man sieht, daß der Verletalgorithmus überlegen ist, wenn man keine sehr genauen Ergebnisse braucht. Die Präzision kann man am besten verbessern, indem man in einem P-K Schema die Ordnung erhöht.
Mit einem Algorithmus 5.Ordnung bestimmt man einen Näherungswert für $y(t+2\Delta t)$. y_1 sei der Wert, den man bei einer Berechnung in einem Schritt erhält. y_2 sei der Wert, den man bei einer Berechnung in zwei Schritten erhält. Dann kann man annehmen:

$$y(t+\Delta t) = \begin{cases} y_1 + & (2\Delta t)^5\Phi + & O(\Delta t^6) \\ y_2 + & 2(\Delta t)^5\Phi + & O(\Delta t^6) \end{cases}$$

Als Fehler definiert man nun

$$\Delta = y_1 - y_2 \quad \propto \quad 30\Delta t^5\Phi \quad .$$

Damit kann man nun eine weitere Annäherung an den wahren Wert von $y(t+2\Delta t)$ berechnen.

$$y(t+2\Delta t) = y_2 + \frac{\Delta}{15} + O(\Delta t^6)$$

Der Fehler Δ kann aber auch dazu verwendet werden, den Zeitschritt Δt adaptiv anzupassen. Wie oben bereits erläutert, ist es erstrebenswert, den Zeitschritt immer so groß wie möglich zu halten. In Bereichen, in denen die Teilchenbahn nur schwach gekrümmt ist, wäre eine kleiner Zeitschritt eine unnötige Verschwendung von Rechenzeit. In Bereichen mit einer starken Krümmung der Teilchenbahn ist eine kleiner Zeitschritt hingegen erforderlich um noch eine hinreichende Genauigkeit zu erhalten. Die Abbildung illustriert dieses Verhalten.

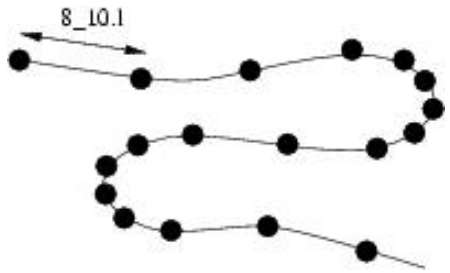


Abb. 8.3 *Anpassung des Zeitschritts*

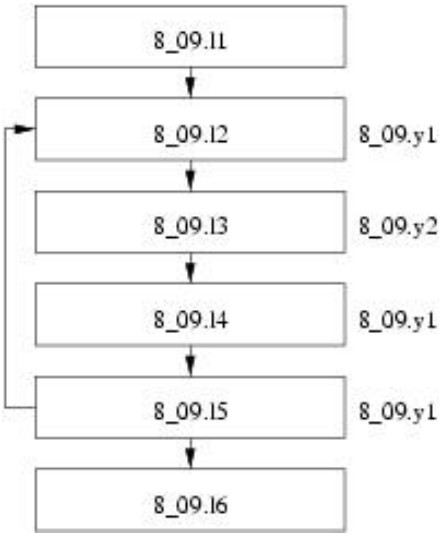
Im Algorithmus verwirklicht man diese Anpassung des Zeitschritts, indem man zunächst einen maximal akzeptablen Fehler $\Delta_{erwartet}$ definiert. In jedem Zeitschritt berechnet man dann die Länge des nächsten Zeitschritts durch

$$\Delta t_{neu} = \Delta t_{alt} \left(\frac{\Delta_{erwartet}}{\Delta_{gemessen}} \right)^{\frac{1}{5}}$$

Der Exponent $\frac{1}{5}$ gilt nur für den besprochenen Algorithmus 5.Ordnung. Für einen Algorithmus n-ter Ordnung erhält man $\frac{1}{n}$.

8.4 Programmiertricks

8.4.1 Allgemeines



Betrachtet man den Aufbau eines Prediktor-Korrektor Programms so erkennt man, daß nur die Kraftberechnung eine doppelte Schleife über N hat, also wie $O(N^2)$ anwächst. Möchte man das Programm schneller machen, muß man an dieser Stelle optimieren. Man kann zum einen die einzelnen Rechnungen optimieren und zum anderen die Länge der Schleife auf $O(N)$ durch Listen verkürzen. Beide Möglichkeiten werden im Folgenden besprochen.

8.4.2 Kraftberechnung

Die Kraft zwischen zwei Teilchen hängt meistens nur vom Abstand der Teilchen ab. In einem naiven Zugang würde man zum Berechnen des Abstandes $r = \sqrt{r_x^2 + r_y^2 + r_z^2}$ die Operation “Wurzel ziehen” ausführen. Im Computer geschieht typischerweise folgendes (entweder in Hard- oder Software):

1. Aus einer 2^{13}Bit großen Tabelle liest der Prozessor einen ersten Näherungswert y_1 heraus, der auf die ersten 13 Bit genau ist. Die Abweichung vom wahren Wert $\sqrt{x}$ sei ε. Dann kann man schreiben:

$$y_1 = \sqrt{x} + \varepsilon \quad \text{mit} \quad \varepsilon \sim O(2^{-14})$$

2. Aus einer zweiten, gleich großen Tabelle holt sich der Computer anschließend einen Näherungswert für $1/\sqrt{x}$. Eine zweite, bessere Approximation der Wurzel ergibt sich dann aus:

$$y_2 = y_1(2 - \frac{1}{\sqrt{x}} y_1) = (\sqrt{x} + \varepsilon)(1 - \frac{\varepsilon}{\sqrt{x}}) = \sqrt{x} - \frac{\varepsilon^2}{\sqrt{x}}$$

Dieses Ergebnis hat nun schon eine Genauigkeit proportional zu 2^{-26}.

3. Der letzte Schritt wird nun solange wiederholt, bis die gewünschte Genauigkeit von z.B. 64 Bits erreicht ist.

Die Berechnung einer Wurzel ist also sehr "teuer". Ähnliches gilt auch für trigonometrische oder transzendente Funktionen wie den Logarithmus. Auf der anderen Seite gibt es "billige" Operationen. Hierzu zählen auf den meisten modernen Prozessoren Integer und Gleitkomma-Operationen wie Addition, Subtraktion, Multiplikation, Bitschiebe- oder logische Operationen. Eine "teure" elementare Operation ist insbesondere die Division (sowohl Gleit- als auch Festkomma). Die exakte Dauer der "schnellen" Operationen hängt bei vielen modernen Prozessoren vom Programmkontext ab und liegt irgendwo zwischen einem und etwa 10-20 Taktzyklen, wobei im schlimmsten Falle prozessorinterne Pipelines geleert und eventuell Operanden aus externen DRAM-Speicher nachgeladen werden müssen.
Ein schneller Algorithmus sollte daher versuchen, die Berechnung von Wurzeln (sowie von Logarithmus- und trigonometrischen Funktionen) zu vermeiden. Man sollte aber auf jeden Fall mit Hilfe eines "profiling" Tools sicherstellen, dass eine vermeintliche Verbesserung wirklich eine solche ist und nicht vielleicht eine Umstellung des Algorithmus andere "teure" Probleme, z.B. bei der Abfolge des Speicherzugriffe, induziert hat.
Meistens sind Potentiale gerade Funktionen, d.h. hängen nur von Termen der Form r^{2n} ab, wie z.B. das harmonische Potential oder das Lenard-Jones Potential. Dann läßt es sich leicht vermeiden, die Wurzel in

$$r = \sqrt{\sum_{\alpha=1}^{dim} (r_i^\alpha - r_j^\alpha)^2}$$

zu berechnen, denn die Kraft hat dann die Form

$$\vec{f}_i = f(r^{2(n-1)})\vec{r}_i \quad .$$

Oft kann man das Potential bei einem "cut-off" Radius r_c abschneiden, so daß es für größere Abstände exakt Null ist. Für ein Lenard-Jones Potential z.B. ist typischerweise $r_c = 2.5\sigma$ ein guter Wert. Wenn das Potential nicht durch einen sehr einfachen Ausdruck beschrieben wird, kann man nun zur weiteren Beschleunigung der Kraftberechnung das komplette Potential tabellieren ("look-up" Tafel). Dazu teilt man das Intervall $(0, r_c^2)$ in K gleich große Stücke, mit den Stützpunkten

$$l_k = \frac{k}{K} r_c^2 \quad .$$

Die Einträge in die Tafel berechnen sich nach

$$F(k) = f(\sqrt{l_k})$$

Die Kraftberechnung reduziert sich dann auf die Berechnung des Indizes k zu einem Teilchenpaar i und j

$$k = [\sum_{\alpha}(r_i^{\alpha} - r_j^{\alpha})(r_i^{\alpha} - r_j^{\alpha})S] + 1 \quad \text{mit} \quad S = \frac{K}{r_c^2}$$

($[\ldots]$ ist die Gaußsche Klammer, d.h. $[x]$ ist die nächste ganze Zahl, die kleiner ist als x.) Das ist die einzige Rechnung, die noch innerhalb der zeitkritischen inneren Schleife auszuführen ist.
Es ist nun noch möglich, den durch die Diskretisierung entstandenen Fehler, durch eine lineare Interpolation zu verkleinern (Newton-Gregory).

$$f(x) = F(k) + (k - xS)\,(F(k-1) - F(k)) \quad \text{mit} \quad x = \sum_{\alpha}(r_i^{\alpha} - r_j^{\alpha})^2$$

8.4.3 Verlettafeln

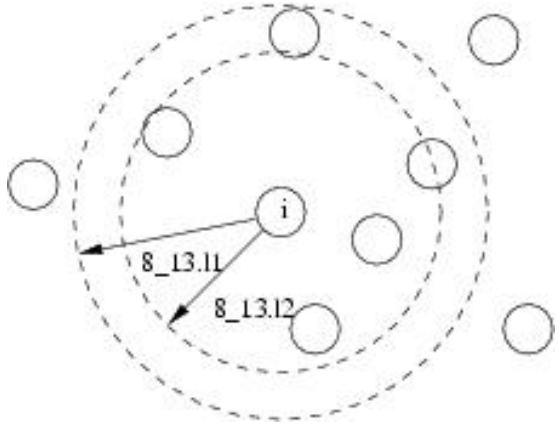


Abb. 8.4 *cut-off Radius und Bereich mit Wechselwirkungpartnern des i-ten Teilchens*

Um nicht so oft alle Teilchenpaare abzufragen, was ja zur $O(N^2)$ Abhängigkeit führt, kann man um jedes Teilchen eine Umgebung vom Radius $r_l > r_c$ betrachten. Die Koordinaten aller Teilchen in der Umgebung werden in der Verlettafel LIST der Länge $N \cdot N_u$ gespeichert, wobei N_u die mittlere Zahl von Teilchen in der Umgebung ist. LIST ist ein eindimensionaler Vektor, in dem hintereinander die Umgebungen aller Teilchen abgespeichert sind. Ein weiterer Vektor POINT[i] gibt den Index des ersten Teilchens der Umgebung des Teilchen i im Vektor LIST an. Die Teilchen in der Umgebung von i sind also in LIST[POINT[i]], ... ,LIST[POINT[$i+1$]-1]. Es genügt also bei der Berechnung der Wechselwirkungen des Teilchens i, nur diese Liste statt aller N Teilchen im System zu durchlaufen, was viel Rechenzeit spart. Allerdings muß die Verlettafel alle $n = (r_l - r_c)/(2\Delta t v_{max})$ (n typischerweise $\approx 10 - 20$) erneuert werden, da in dieser Zeit andere Teilchen von außerhalb r_l in den Wechselwirkungsradius r_c eintreten können. Dabei ist v_{max} die schnellste im System auftretende Geschwindigkeit. Das

Auffrischen der Tafel erfordert wieder $\propto N^2$ Operationen. Dadurch wächst auch der Algorithmus global immer noch wie N^2.

8.4.4 Zellmethoden

Man legt über das System ein hyperkubisches Netz der Größe M^d, so daß jede einzelne Zelle länger als $2r_c$ ist. (d ist die Dimension.) Im zweidimensionalen Fall liegen die in Frage kommenden Wechselwirkungspartner des Teilchens i dann in einer der neun in der Abbildung dunkel unterlegten Zellen.

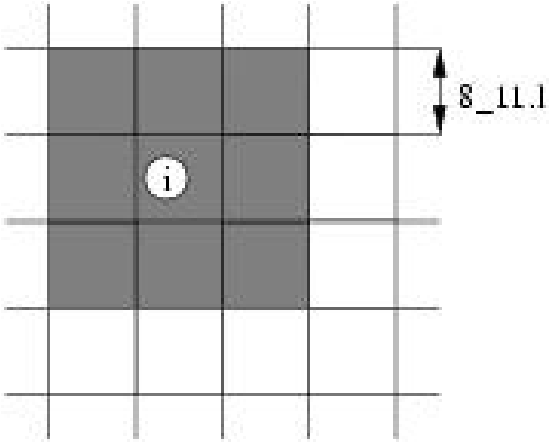


Abb. 8.5 *Zellen mit Wechselwirkungspartnern des i-ten Teilchens*

Man braucht im Mittel also nur $3^d N^2/M^d$ Teilchen zu prüfen, so daß sich die Schleife um $(M/3)^d$ verkürzt.
Um zu wissen, welche Teilchen in einer Zelle sind, benutzt man am besten "linked-cells" (Knuth, 1973): In einem Vektor ANF der Länge M^d speichert man für jede Zelle die Addresse des ersten Teilchens in der Zelle. Enthält die Zelle kein Teilchen, setzt man ANF auf Null. In einem Vektor LIST der Länge N speichert man für jedes Teilchen die Addresse des nächsten Teilchens in der Zelle. Für das letzte Teilchen in der Zelle trägt man Null ein.

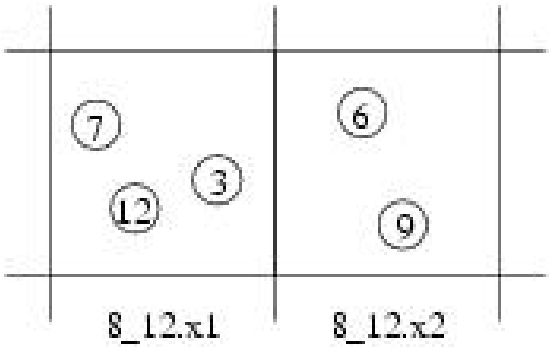


Abb. 8.6 *Zwei Zellen einer linked-cell Struktur*

Für das in der Abbildung gezeigte Beispiel hätte man also folgende Einträge in ANF und LIST:

ANF[1] = 7	LIST[7] = 3
	LIST[3] = 12
	LIST[12] = 0
ANF[2] = 6	LIST[6] = 9
	LIST[9] = 0

Es ist einfach zu bestimmen, in welcher Zelle ein bestimmtes Teilchen liegt (z.B. durch Trunkation). Deshalb kann man, wenn ein Teilchen in eine benachbarte Zelle fliegt, die Vektoren ANF und LIST direkt erneuern, so daß keine Schleife über alle Teilchen N mehr nötig ist. Der Algorithmus geht dann echt wie N in der CPU Zeit.

Man kann auch die Zellen kleiner als $2r_c$ wählen, was die Listen erspart, jedoch viel Speicherplatz kosten kann.

8.4.5 Parallelisierung

Am besten kann man ein Programm parallelisieren, indem man die linked-cell Struktur in Scheiben zerschneidet, so daß in jedem Segment noch viele linked-cell Zellen sind:

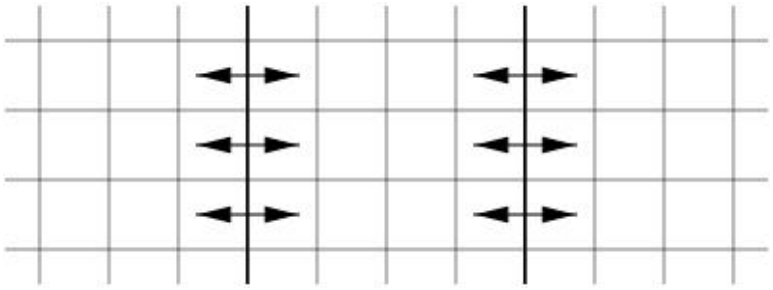


Abb. 8.7 *Parallelisierung und message-passing*

Jedes Segment wird von einem anderen Prozessor behandelt. Für die Kraftberechnung wird von jeder Zelle die linke Spalte von linked-cell Zellen mit “message-passing” auf den rechts daneben liegenden Prozessor geschickt. Ebenfalls message-passing benutzt man, wenn man die linked-cell Struktur erneuert und ein Teilchen von einem Prozessor zum anderen geflogen ist. Ein weiterer Prozessor kümmert sich um die Auswertung der Daten und bekommt diese ebenfalls durch message-passing. Man kann die Segmente auch variabel lang machen und zwar so, daß ca gleich viele Teilchen in jedem Segment sind. Durch dies “load-balancing” werden die Prozessoren gleich stark belastet, was die Parallelisierung effizienter macht.

8.5 Langreichweitige Kräfte

8.5.1 Allgemeines

Schneidet man ein Potential bei r_c ab, wird der Potentialverlauf an dieser Stelle unstetig - das Potential macht einen Sprung. Das führt dazu, daß die Kraft an dieser Stelle unendlich wird. Die Molekulardynamik versagt an dieser Stelle und die Energieerhaltung im System ist nicht mehr gewährleistet. Um das zu verhindern, muß man das Potential so verschieben, daß $\widetilde{\mathcal{V}}(r_c) = 0$ gilt, d.h.

$$\widetilde{\mathcal{V}}(r) = \begin{cases} \mathcal{V}(r) - \mathcal{V}_c & \text{für} \quad r \leq r_c \quad \text{mit} \quad \mathcal{V}_c = \mathcal{V}(r_c) \\ 0 & \text{für} \quad r > r_c \end{cases}$$

Damit erreicht man, daß die Kraft an dieser Stelle endlich bleibt. Man hat jedoch weiterhin einen Sprung im Kraftverlauf. Um auch diesen noch zu glätten, führt man

$$\widetilde{\widetilde{\mathcal{V}}}(r) = \begin{cases} \mathcal{V}(r) - \mathcal{V}_c - \frac{\partial \mathcal{V}}{\partial r}|_{r_c}(r - r_c) & r \leq r_c \\ 0 & r > r_c \end{cases}$$

ein. Mit diesem letzten Potential ist nun auch die Energiererhaltung gewährleistet. Allerdings hat man jetzt eine leicht geänderte Physik.
Als langreichweitige Kräfte bezeichnet man Kräfte, die nicht schneller als r^{-d} abfallen, wobei d die Dimension des Systems ist. Hierzu zählen z.B. die Gravitationskraft und die elektrostatische Wechselwirkung. Wie bereits oben erwähnt entsteht bei der Behandlung solcher Kräfte ein Problem dadurch, daß die Reichweite dieser Kräfte größer ist als die Ausdehnung des simulierten Systems. Eine einfache Vergrößerung des Systems ist nicht möglich, da sie unpraktikabel lange Rechenzeiten erfordern würden. Auch kann man keinen cut-off r_c einführen. Im Falle der elektrostatischen Wechselwirkung wäre ein Abschneiden des Potentials bei r_c gleichbedeutend mit einer geladenen Kugeloberfläche im Abstand r_c um das betrachtete Ion. Man kann diese Ladung durch eine entgegengesetzte Ladung auf der Kugelfläche kompensieren. Das entspricht aber einem Verschieben des Potentials und damit einer Änderung der Physik.

8.5.2 Die Ewaldsumme

Unter Benutzung periodischer Randbedingungen kann man die Kräfte mit der Ewaldsumme (1921) berechnen. Dieses Verfahren wurde ursprünglich für Ionenkristalle entwickelt. Man betrachtet dabei auch die Wechselwirkungen mit den periodisch fortgesetzten Kopien des zentralen Kastens.

$$\mathcal{V} = \frac{1}{2}\sum_{\vec{n}}{}' \sum_{i,j}^{N} z_i z_j |\vec{r}_{ij} + \vec{n}|^{-1}$$

z_i, z_j sind die Ladungen. $\vec{n}$ ist der Vektor, der vom Zentrum des ursprünglichen Systems auf das Zentrum jedes Spiegelbildes zeigt. Der Strich an der ersten Summe bedeutet, daß der Fall $j = i \wedge \vec{n} = 0$ ausgelassen wird.
Jede Summation über $\vec{n}$ entspricht der Annäherung an ein unendliches System durch die Hinzunahme einer weiteren "Kugelschale" aus Spiegelbildern.

8.5.3 Aufweichen des Potentials

Man führt Ladungsverteilungen um die gegebenen Punktladungen ein, die diese abschirmen und dadurch kurzreichweitig machen. Üblicherweise nimmt man an, die Ladungsverteilung der Abschirmladungen habe Gaußsche Form.

$$\rho_i(\vec{r}) = \frac{z_i \kappa^3}{\sqrt{\pi^3}} e^{-\kappa^2 (\vec{r}-\vec{r}_i)^2}$$

κ ist ein frei wählbarer Parameter, der die Breite der Ladungsverteilung bestimmt. Er wird an die Erfordernisse des Algorithmuses angepaßt. Typischerweise wählt man κ so, daß die Breite der Gaußkurve in der Größenordnung der Kastenbreite liegt, d.h $\kappa \approx 5/L$.

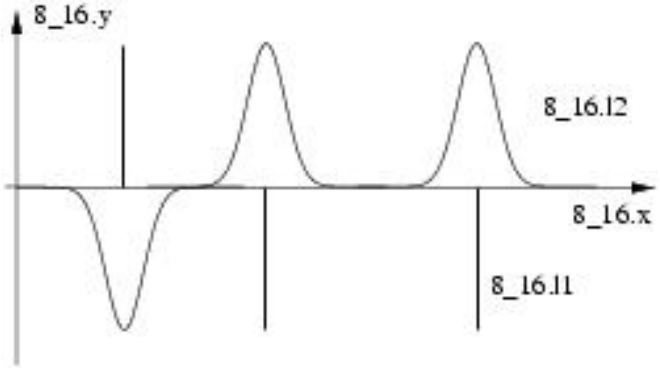


Abb. 8.8 *Ladungsverteilung*

Man kann das Potential aus Punktladungen und Abschirmladungen nun aufsummieren und erhält:

$$\mathcal{V}_1 = \frac{1}{2} \sum_{i,j}^{N} \sum_{\vec{n}=0}^{\prime} z_i z_j \frac{\mathrm{erfc}(\kappa |\vec{r}_{ij} + \vec{n}|)}{|\vec{r}_{ij} + \vec{n}|}$$

erfc ist die *error-function* ($\mathrm{erfc}(x) = 2/\sqrt{\pi} \int_x^\infty \exp(-y^2) dy$). Die oben eingeführten Abschirmladungen muß man nun wieder abziehen. Dazu addiert man noch einmal dieselbe Ladungsverteilung jedoch mit umgekehrtem Vorzeichen. Die Aufsummation von Gaußkurven am Ort jeder Punktladung führt man am einfachsten als Faltungsprodukt im k-Raum aus und addiert dann die inverse Fouriertransformation davon.

$$\mathcal{V}_2 = \frac{1}{2\pi L^3} \sum_{i,j}^{N} \sum_{k \neq 0} \frac{4\pi^2 z_i z_j}{k^2} \exp(-\frac{k^2}{4\kappa^2}) \cos(\vec{k}\vec{r}_{ij})$$

Als letzte Korrektur muß man nun noch die Wechselwirkung eines Teilchens am Ort $\vec{r}_i$ mit sich selbst wieder abziehen.

$$\mathcal{V}_3 = -\frac{\kappa}{\sqrt{\pi}} \sum_{i=1}^{N} z_i^2$$

Das komplette Potential ergibt sich aus der Summe der Terme für $\mathcal{V}_1, \mathcal{V}_2$ und $\mathcal{V}_3$.

8.5.4 Reaktionsfeldmethode

In der Reaktionsfeldmethode teilt man die Umgebung (und damit auch die Wechselwirkungen) des Teilchens *i* in zwei Anteile.

1. Direkte Dipol-Dipol-Wechselwirkungen mit den Teilchen innerhalb einer kugelförmigen Zelle mit Radius r_c.

2. Wechselwirkung mit dem Bereich außerhalb der Zelle, der als dielektrisches Kontinuum mit der Dielektizitätskonstanten ε_s aufgefaßt wird.

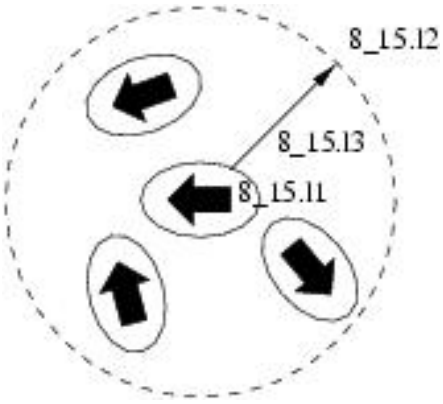


Abb. 8.9 *Reaktionsfeldmethode*

Das Kontinuum außerhalb der Zelle wird durch die Dipole innerhalb des Bereichs polarisiert. Das resultierende elektrische Feld

$$E_i = \frac{2(\varepsilon_s - 1)}{2\varepsilon_s + 1} \frac{1}{r_c^3} \sum_{j \in r_c} \mu_j \quad ,$$

das sogenannte Reaktionsfeld, wirkt nun auf das Teilchen *i* zurück. Die Kraft auf das Teilchen *i* setzt sich somit zusammen aus

$$\vec{f}_i = \sum_{j \in r_c} \vec{f}_{ij} + E_i \times \mu_i \quad .$$

Der erste Term ist die Dipol-Dipol-Wechselwirkung mit den Teilchen innerhalb von r_c, der zweite Term ist die Wechselwirkung mit dem umgebenden Kontinuum.

In der angegebenen Form wird immer dann ein Sprung im Verlauf der Kraft auftreten, wenn ein Teilchen die Zelle verläßt oder neu in sie eintritt. Um das zu verhindern schwächt man durch eine Wichtungsfunktion g die Wirkung des Teilchens ab, je näher es dem Rand der Zelle kommt. Eine mögliche Wahl der Wichtungsfunktion ist

$$g(r_j) = \begin{cases} 1 & r_j < r_t \\ \frac{r_j - r_c}{r_c - r_t} & r_t \leq r_j \leq r_c \\ 0 & r_j > r_c \end{cases} \quad .$$

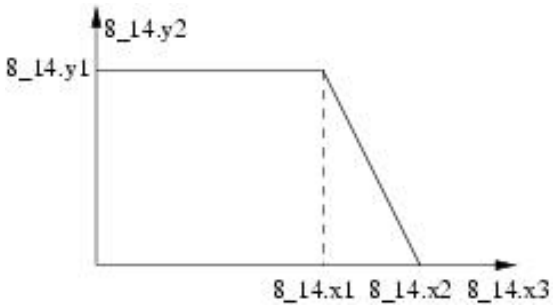


Abb. 8.10 *Wichtungsfunktion*

8.6 Moleküle

8.6.1 Einführung

Bei der Simulation von Molekülen und Polymeren muß man Körper behandeln, die aus N chemisch gebundenen Teilchen aufgebaut sind. Es wäre nun möglich, die chemische Bindung durch Potentiale zu beschreiben. Das ist in der Regel jedoch zu aufwendig. Zudem sind die Kräfte zwischen den Atomen zumeist wesentlich größer als die Wechselwirkungen zwischen den Molekülen. Man kann daher annehmen, die Bindungen seien starr und die Bindungswinkel fest vorgegeben. Um diese Nebenbedingungen zu erfüllen, gibt es zwei unterschiedliche Verfahren.

1. Man verwendet die Methode der Lagrangeschen Multiplikatoren. Hier hat man ein Gleichungssystem mit k Nebenbedingungen zu lösen.

2. Man behandelt das komplette Molekül als starren Körper.

Beide Methoden werden im Folgenden kurz vorgestellt.

8.6.2 Methode der Lagrangeschen Multiplikatoren

Als Beispiel sei ein Wassermolekül behandelt. Die Bewegungsgleichungen lauten:

$$m\ddot{\vec{r}}_i = \vec{f}_i + \vec{g}_i$$

$\vec{f}_i$ seien die auf das i-te Teilchen wirkenden Kräfte. Die Kräfte $\vec{g}_i$ soll die Erfüllung der Nebenbedingungen gewährleisten. Die Nebenbedingungen, nämlich konstante, mittlere Atomabstände, sind gegeben durch:

$$\begin{aligned} \chi_{12} &= r_{12}^2 - d_{12}^2 = 0 \\ \chi_{23} &= r_{23}^2 - d_{23}^2 = 0 \end{aligned}$$

r_{ij} sei dabei definiert als

$$r_{ij} = \sqrt{\sum_k \left(r_i^{(k)} - r_j^{(k)} \right)^2}$$

Die Funktion $\vec{g}_i$ lautet dann:

$$\begin{aligned} \vec{g}_i &= \frac{1}{2}\lambda_{12}\vec{\nabla}_{r_i}\chi_{12} + \frac{1}{2}\lambda_{23}\vec{\nabla}_{r_i}\chi_{23} \\ \Rightarrow \quad \vec{g}_1 &= \lambda_{12}\vec{r}_{12}, \qquad \vec{g}_2 = \lambda_{23}\vec{r}_{23} - \lambda_{12}\vec{r}_{12}, \qquad \vec{g}_3 = -\lambda_{23}\vec{r}_{23} \end{aligned}$$

Die "Federkonstanten" λ_{ij} hängen vom Lösungsverfahren ab. Z.B. erfordert ein großer Zeitschritt Δt des Lösungsverfahrens starke Federn, d.h. große λ_{ij}.
Eingesetzt in einen Verlet-Algorithmus erhält man nun

$$\begin{aligned} \vec{r}_i(t+\Delta t) &= 2\vec{r}_i(t) - \vec{r}_i(t-\Delta t) + \Delta t^2 \frac{\vec{f}_i}{m_i} + \Delta t^2 \frac{\vec{g}_i}{m_i} \\ &= \vec{r}'_i(t+\Delta t) + \Delta t^2 \frac{\vec{g}_i}{m_i} \end{aligned}$$

bzw.

$$\begin{aligned} \vec{r}_1(t+\Delta t) &= \vec{r}'_1(t+\Delta t) + \Delta t^2 \frac{\lambda_{12}}{m_1}\vec{r}_{12}(t) \\ \vec{r}_2(t+\Delta t) &= \vec{r}'_2(t+\Delta t) + \Delta t^2 \frac{\lambda_{23}}{m_2}\vec{r}_{23}(t) - \Delta t^2 \frac{\lambda_{12}}{m_2}\vec{r}_{12}(t) \\ \vec{r}_3(t+\Delta t) &= \vec{r}'_3(t+\Delta t) - \Delta t^2 \frac{\lambda_{23}}{m_3}\vec{r}_{23}(t) \end{aligned}$$

Die λ_{ij} berechnet man aus den Gleichungen für die Nebenbedingungen.

$$\begin{aligned} [\vec{r}'_1(t+\Delta t) - \vec{r}'_2(t+\Delta t) + \Delta t^2\lambda_{12}(\frac{1}{m_1} + \frac{1}{m_2})\vec{r}_{12}(t) - \Delta t^2\lambda_{23}\frac{\vec{r}_{23}(t)}{m_2}]^2 &= d_{12}^2 \\ [\vec{r}'_2(t+\Delta t) - \vec{r}'_3(t+\Delta t) + \Delta t^2\lambda_{23}(\frac{1}{m_2} + \frac{1}{m_3})\vec{r}_{23}(t) - \Delta t^2\lambda_{12}\frac{\vec{r}_{12}(t)}{m_2}]^2 &= d_{23}^2 \end{aligned}$$

$(\frac{1}{m_1} + \frac{1}{m_2})$ ist die reduzierte Masse. Dieses Gleichungssystem für λ_{12} und λ_{23} muß man für jeden Zeitschritt erneut lösen.
Die Erhaltung des Bindungswinkels behandelt man analog, indem man eine zusätzliche Bindung einführt, die den Winkel konstant hält.

8.6.3 Starre Moleküle

Man betrachtet ein Molekül als starren Körper mit 3 Freiheitsgraden in 2 Dimensionen und 6 Freiheitsgraden in 3 Dimensionen. Die Bindungslängen und -winkel werden nun streng festgehalten.
Die Gesamtbewegung eines Moleküls zerlegt man in die Translationsbewegung des Schwerpunktes und die Rotationsbewegung in einem Koordinatensystem mit dem Schwerpunkt als Ursprung. Der Schwerpunkt $\vec{r}_s$ eines aus N Teilchen aufgebauten starren Körpers ist definiert als

$$M\vec{r}_s = \sum_i^N \vec{r}_i m_i \qquad .$$

M ist die Gesamtmasse des Körpers, $\vec{r}_s$ der Ortsvektor des Schwerpunktes. Im Folgenden sei $\vec{d}_i = \vec{r}_i - \vec{r}_s$ der Abstandsvektor des Teilchens i vom Schwerpunkt. Der Körper bewegt sich nun translatorisch so, als griffen alle auf den Körper wirkenden Kräfte an seinem Schwerpunkt an.

$$M\ddot{\vec{r}}_s = \vec{f}_s \quad \text{mit} \quad \vec{f}_s = \sum_i^N \vec{f}_i$$

Diese Gleichung kann man nun mit der Verlet-Methode oder einem Prediktor-Korrektor Verfahren lösen. In zwei Dimensionen hat man nur einen Drehfreiheitsgrad ϕ der durch die Definitionsgleichungen für die Winkelgeschwindigkeit $\omega = \dot{\phi}$, das Drehmoment $\tau = \sum_i^N \vec{d}_i \times \vec{f}_i$, das Trägheitsmoment $I = \sum_i^N d_i^2 m_i$ und die Bewegungsgleichung

$$I\dot{\omega} = \tau$$

charakterisiert wird.
Die Behandlung der Drehbewegung ist schwieriger in drei Dimensionen. Als Drehimpuls des starren Körpers definiert man

$$\begin{aligned}
\vec{l} &= \sum_i^N m_i \vec{d}_i \times \vec{v}_i \\
&= \sum_i^N m_i \vec{d}_i \times (\vec{d}_i \times \vec{\omega}) \\
&= \sum_i^N m_i (\vec{d}_i (\vec{d}_i \vec{\omega}) - \vec{d}_i^2 \vec{\omega}) \\
&= \mathbf{I}\vec{\omega} \qquad .
\end{aligned}$$

In der letzten Zeile wurde dabei der Trägheitstensor $\mathbf{I}$ eingeführt. Mit Hilfe des dyadischen Produkts kann man den Trägheitstensor auch schreiben als

$$\mathbf{I} = \sum_i^N m_i (\vec{d}_i^{\,t} \otimes \vec{d}_i - \vec{d}_i^2 \mathbf{1}) \qquad .$$

Das Drehmoment auf den starren Körper berechnet sich nach

$$\vec{\tau} = \sum_i^N \vec{d}_i \times \vec{f}_i \qquad .$$

Damit hat man als Bewegungsgleichung für die Drehbewegung

$$\dot{\vec{l}} = \vec{\tau} \qquad .$$

Der Drehimpuls und das Drehmoment sind dabei in Bezug auf den Körperschwerpunkt anzugeben.
Durch die Eigenvektorrichtungen, die Hauptträgheitsachsen, von **I** und den Schwerpunkt als Ursprung wird ein körperfestes Koordinatensystem vorgegeben. In diesem Koordinatensystem sind die $\vec{d}_i$ und **I** konstant. Man löst die Bewegungsgleichung

$$\frac{d}{dt}|_{\text{Labor}}\vec{l} = \frac{d}{dt}|_{\text{Körper}}\vec{l} + \vec{\omega} \times \vec{l} = \vec{\tau}$$

daher am einfachsten in diesem System. Zur Transformation vom mitbewegten, körperfesten System auf das Laborsystem verwendet man im Allgemeinen die *Eulerschen Winkel*. Die Eulerschen Winkel sind wie folgt definiert:

1. Drehung um die *z*-Achse mit dem Winkel Φ und der Winkelgeschwindigkeit

$$\vec{\omega}_\Phi = \begin{pmatrix} 0 \\ 0 \\ \dot{\Phi} \end{pmatrix} \qquad .$$

2. Anschließende Drehung um die neu entstanden x-Achse um den Winkel Θ mit der Winkelgeschwindigkeit

$$\vec{\omega}_\Theta = \begin{pmatrix} \cos\Phi & -\sin\Phi & 0 \\ \sin\Phi & \cos\Phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{\Theta} \\ 0 \\ 0 \end{pmatrix} \qquad .$$

   Die Winkelgeschwindigkeit ist hier bereits im ursprünglichen, laborfesten Koordinatensystem ausgedrückt.

3. Drehung um die neue *z*-Achse um den Winkel Ψ mit der Winkelgeschwindigkeit

$$\vec{\omega}_\Psi = \begin{pmatrix} \cos\Phi & -\sin\Phi & 0 \\ \sin\Phi & \cos\Phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\Theta & -\sin\Theta \\ 0 & \sin\Theta & \cos\Theta \end{pmatrix} \begin{pmatrix} 0 \\ \dot{\Psi} \\ 0 \end{pmatrix} \qquad .$$

   Wiederum wurde die Winkelgeschwindigkeit im ursprünglichen System ausgedrückt.

Die Reihenfolge der Drehungen ist vorgegeben. Eine andere Reihenfolge liefert eine anderes Koordinatensystem. Für weitere Erläuterungen siehe z.B. [16].
Die drei obigen Terme liefern die komplette Winkelgeschwindigkeit im Laborsystem. Aufgelöst nach $\dot{\Psi}, \dot{\Theta}$ und $\dot{\Phi}$ erhält man die Bewegungsgleichungen für die Eulerschen Winkel:

$$\begin{aligned}
\dot{\Phi} &= -\omega_x \frac{\sin\Phi\cos\Theta}{\sin\Theta} + \omega_y \frac{\cos\Phi\cos\Theta}{\sin\Theta} + \omega_z \\
\dot{\Theta} &= \omega_x \cos\Theta + \omega_y \sin\Phi \\
\dot{\Psi} &= \omega_x \frac{\sin\Phi}{\sin\Theta} - \omega_y \frac{cos\Phi}{\sin\Theta}
\end{aligned}$$

Man kann nun die Bewegungsgleichung im körperfesten System lösen, die erhaltene Winkelgeschwindigkeit ins Laborsystem transformieren und dort mit den zuletzt angegebene Gleichungen die neuen Eulerschen Winkel berechnen. Diese Methode hat nun das Problem, daß die Bewegungsgleichungen der Eulerschen Winkel singulär werden, sobald $\Theta = 0$ oder π wird. Dieses Problem umgeht man mit der Einführung von *Quaternionen*.
Als Quaternion bezeichnet man die Größe

$$Q = (q_0, q_1, q_2, q_3) \qquad ,$$

mit

$$\begin{aligned}
q_0 &= \cos\frac{1}{2}\Theta\cos\frac{1}{2}(\Phi+\Psi) \\
q_1 &= \sin\frac{1}{2}\Theta\cos\frac{1}{2}(\Phi-\Psi) \\
q_2 &= \sin\frac{1}{2}\Theta\sin\frac{1}{2}(\Phi-\Psi) \\
q_3 &= \cos\frac{1}{2}\Theta\sin\frac{1}{2}(\Phi+\Psi) \quad \text{mit} \quad \sum_0^3 q_j^2 = 1 \qquad .
\end{aligned}$$

Als Bewegungsgleichung eines Quaternions erhält man

$$\begin{pmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix} \begin{pmatrix} 0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \qquad .$$

Die ω_i sind die Komponenten der Winkelgeschwindigkeit im körperfesten System. Statt der obigen Bewegungsgleichungen der Eulerschen Winkel hat man nun also ein System von gekoppelten Differentialgleichungen erster Ordnung, das keine störenden Singularitäten mehr enthält. Auch die Transformationsmatrix vom Laborsystem in das körperfeste System wird nun natürlich mit Hilfe von Quaternionen formuliert.

In jedem Zeitschritt berechnet man nun die Kräfte auf den Körper im körperfesten System und erhält daraus die Winkelgeschwindigkeit im körperfesten System. Damit berechnet man das neue Quaternion des Körpers und daraus schließlich die Eulerschen Winkel, d.h. die neue Lage.

8.7 Molekulardynamik bei konstanter Temperatur

8.7.1 Vorbemerkungen

In den bisherigen Simulationen wurde die Energie, das Volumen und die Teilchenzahl konstant gehalten, d.h. wir behandelten mikrokanonische NEV-Ensembles. Die Mitglieder eines solchen Ensembles liegen im Konfigurationsraum alle auf einer Fläche konstanter Energie. Die Verteilungsfunktion ist deltaförmig

$$f(\vec{p},\vec{q}) \sim \delta(\mathcal{H}(\vec{p},\vec{q}) - E_0)$$

Derartige Simulationen entsprechen nun oft nicht der physikalischen Wirklichkeit. Häufig ist es sinnvoller, mit konstant gehaltener Temperatur (kanonisches NTV-Ensemble) und konstant gehaltenem Druck, d.h. im sogenannten kanonischen pNT-Ensemble zu rechnen. Anschaulich bedeutet dies, daß man das System an ein Wärmebad ankoppelt bzw. in einen Kasten einschließt, der mit einem beweglichen Stempel abgeschlossen wird. Für die Verteilungsfunktion eines kanonischen Ensembles gilt

$$f(\vec{p},\vec{q}) \sim \frac{1}{Z}\exp(-\frac{\mathcal{H}(\vec{p},\vec{q})}{kT})$$

Um bei konstanter Temperatur zu simulieren, ist es zunächst erforderlich, den Begriff der Temperatur mikroskopisch zu definieren. Der Gleichverteilungssatz besagt, daß jeder Freiheitsgrad, der homogen quadratisch in der Hamiltonfunktion auftaucht, im Mittel die Energie $\frac{1}{2}kT$ besitzt. (Eine Funktion heißt *homogen n-ten Grades*, falls gilt: $f(\lambda x) = \lambda^n f(x)$.) Die mathematische Formulierung des verallgemeinerten Gleichverteilungssatzes lautet

$$\langle p_i \frac{\partial \mathcal{H}}{\partial p_i}\rangle = kT \qquad \langle q_i \frac{\partial \mathcal{H}}{\partial q_i}\rangle = kT \qquad .$$

Mit der ersten Formel und $\mathcal{H} = \sum_i \frac{p_i^2}{2m_i} + V(\vec{r})$ ($V(\vec{r})$ ist eine beliebige Funktion des Teilchenortes) gilt beispielsweise

$$\langle p_i \frac{2p_i}{2m_i}\rangle = 2\frac{1}{2m_i}\langle p_i^2\rangle = 2E_{kin,i} = kT \qquad .$$

Damit kann man nun die instantane Temperatur $\mathcal{T}$ definieren als

$$\mathcal{T} = \frac{2}{k(3N-3)}\sum_j^N \frac{\vec{p}_j^2}{2m_i} \qquad .$$

Der Nenner $3N-3$ ergibt sich aus den $3N$ Freiheitsgraden eines N-Teilchensystems abzüglich der drei Zwangsbedingungen, welche die Erhaltung des Gesamtimpulsvektors gewährleisten sollen.
Im Folgenden werden nun mehrere Methoden zur Erhaltung der Temperatur in einer Molekulardynamik vorgestellt.

8.7.2 Geschwindigkeitsskalierung

In jedem Zeitschritt führt man eine Neuskalierung der Geschwindigkeiten durch:

$$v_i \longrightarrow \alpha v_i.$$

Die Temperatur ändert sich dabei gemäß

$$\mathcal{T} \longrightarrow \alpha^2 \mathcal{T}.$$

Um die gewüschte Tempertur T zu erreichen, wählt man also

$$\alpha = \sqrt{\frac{T}{\mathcal{T}}}.$$

Diese Methode hat natürlich das Problem, daß sich die Teilchen nicht mehr auf den durch die ursprünglichen Bewegungsgleichungen beschriebenen Trajektorien bewegen. Dadurch werden alle dynamischen Informationen, die man dem System evtl. entnehmen möchte, wie z.B. Zeitkorrelationsfunktionen, response-Funktionen, etc. verfälscht.
Man kann die Methode verbessern, indem man eine "sanftere" Skalierung durchführt. Als Skalierungsfaktor wählt man dabei

$$\alpha = (1 + \frac{\Delta t}{t_T}(\frac{T}{\mathcal{T}} - 1))^{\frac{1}{2}}.$$

Δt ist der Zeitschritt der Simulation und t_T eine zuvor bestimmte Zeitkonstante. Auch mit dieser Skalierung hat man jedoch das Problem, das die Trajektorien leicht gestört werden. Auch hat man keine kanonische Verteilung im Phasenraum. Die Methode ist jedoch gut geeignet, um in der Startphase einer Simulation einen Gleichgewichtszustand mit vorgegebener Temperatur einzustellen.

8.7.3 Hoover (1982)

Man führt einen additiven "Reibungsterm" in die Bewegungsgleichungen ein:

$$\dot{\vec{r}} = \frac{\vec{p}}{m}, \qquad \dot{\vec{p}} = \vec{f} - \xi \vec{p}$$

ξ bestimmt man aus

$$\dot{\xi} = \frac{fk}{Q}(\mathcal{T} - T).$$

f ist die Anzahl der Freiheitsgrade. Q ist die *thermische Trägheit*. Sie kontrolliert, wie schnell sich eine Temperaturabweichung auf das System auswirkt. Die so erzeugten Zustände haben nun die gewünschte Verteilung des kanonischen Ensembles. Jedoch sind natürlich auch hier die Trajektorien leicht geändert, was zu Abweichungen bei der Bestimmung von zeitabhängigen Größen führen kann.
Durch die Wahl

$$\xi = \frac{1}{2t_T \mathcal{T}}(\mathcal{T} - T)$$

kann man die im vorigen Kapitel vorgestellte Methode auf eine ähnliche Form bringen. Für genügend kleine Δt, $(\mathcal{T} - T)$ entspricht diese Form nämlich gerade einer Taylorentwicklung des zuletzt angegebenen Terms für α. Man beachte das Fehlen der Zeitableitung des "Reibungskoeffizienten" ξ.

8.7.4 Nosé (1984)

Das Verfahren von Hoover läßt sich mit der hier vorgestellten Idee besser verstehen. Man führt einen weiteren Freiheitsgrad s ein, der als das umgebende Wärmebad angesehen werden kann. Das "Wärmebad" kann Energie aufnehmen und Energie an das System abgeben. Als potentielle bzw. kinetische Energie von s führt man ein

$$\begin{aligned} \mathcal{V}(s) &= (f+1)kT \ln s \\ \mathcal{K}(s) &= \frac{1}{2}Q\dot{s}^2 \end{aligned}$$

Q ist wieder die thermische Trägheit, die den Energiefluß zwischen System und "Wärmebad" regelt. Mit

$$\vec{v} = s\dot{\vec{r}} = s\frac{\vec{p}}{m}$$

ist s an die Geschwindigkeit eines Teilchens gekoppelt. Man kann damit eine neue Lagrangefunktion

$$\mathcal{L} = \frac{1}{2}Q\dot{s}^2 - \mathcal{V}(s) + \sum_i \left[\frac{1}{2}m_i\vec{v}_i^2 - \mathcal{V}(\vec{r}_i)\right]$$

aufstellen. Daraus leitet man die neuen Bewegungsgleichungen mittels

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{s}} - \frac{\partial \mathcal{L}}{\partial s} = 0, \qquad \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\vec{r}}_i} - \frac{\partial \mathcal{L}}{\partial \vec{r}_i} = 0$$

ab:

$$\begin{aligned} &2ms\dot{s}\dot{\vec{r}}_i + ms^2\ddot{\vec{r}}_i - \vec{f}_i = 0, \\ &Q\ddot{s} + \frac{1}{s}(f+1)kT + \sum_i ms\dot{\vec{r}}_i{}^2 = 0. \end{aligned}$$

8.7.5 Stochastische Methode (Andersen, 1980)

Man kann ein kanonisches Ensemble auch durch die Ankopplung eines Monte Carlo Verfahrens simulieren. Dazu wählt man alle m Schritte ein Teilchen aus und gibt ihm einen neuen, nach der Maxwell-Boltzmann Verteilung

$$P(\vec{p}) = \frac{3}{(2\pi mkT)^{\frac{3}{2}}} \exp(-\frac{(\vec{p}-\vec{p}_0)^2}{2mkT})$$

ausgewählten Impuls $\vec{p}$. Für kleines m ist die Kopplung zum Wärmebad schwach, die Energiefluktuationen sind langsam, die Molekulardynamik dominiert und die kanonische Verteilung wird langsam exploriert. Für großes m dominiert die Monte-Carlo Methode, der Konfigurationsraum der Trajektorien wird langsam exploriert und die Temperaturfluktuationen werden durch die Kollisionen bestimmt. Nachteil dieser Methode ist, daß man m im wesentlichen empirisch wählt.
Alternativ kann man auch in größeren Intervallen die Geschwindigkeit aller Teilchen ändern.

8.8 Molekulardynamik bei konstanten Druck

8.8.1 Vorbemerkungen

Die Methoden zur Beschreibung von Systemen mit konstantem Druck sind analog zu den vorgestellten Methoden zur Molekulardynamik mit konstanter Temperatur. Auch hier muß man zunächst wieder einen instantanen Druck definieren. Dazu geht man vom verallgemeinerten Gleichverteilungstheorem aus (siehe oben).

$$\langle q_k \frac{\partial \mathcal{H}}{\partial q_k} \rangle = \langle p_k \frac{\partial \mathcal{H}}{\partial p_k} \rangle = kT.$$

Mit $\mathcal{H} = \mathcal{K}(\vec{p}) + \mathcal{V}(\vec{r})$ erhält man damit

$$\begin{aligned} -\frac{1}{3}\langle \sum_i^N \vec{r}_i \cdot (\vec{\nabla}_{\vec{r}_i} \mathcal{V}(\vec{r})) \rangle &= -NkT \\ \Longrightarrow \frac{1}{3}\langle \sum_i^N \vec{r}_i \cdot (\vec{f}_i^{ext} + \vec{f}_i^{part}) \rangle &= -NkT \\ \Longrightarrow \frac{1}{3}\langle \sum_i^N \vec{r}_i \cdot \vec{f}_i^{ext} \rangle + \frac{1}{3}\langle \sum_i^N \vec{r}_i \cdot \vec{f}_i^{part} \rangle &= -NkT \quad . \end{aligned}$$

$\vec{f}_i^{part}$ ist die Kraft aufgrund der Teilchen-Teilchen-Wechselwirkung. $\vec{f}_i^{ext}$ ist die Kraft der Wände auf die Teilchen. Zur Abkürzung des zweiten Ausdrucks führt man das Virial

$$W = \frac{1}{3}\sum_i^N \vec{r}_i \cdot \vec{f}_i^{part}$$

ein. Mit Hilfe des Drucks auf die Wände des Systems schreibt man für den ersten Term

$$\begin{aligned}
&\frac{1}{3}\langle\sum_i^N \vec{r}_i \vec{f}_i^{ext}\rangle \\
&= -\frac{1}{3}\int \vec{r} p d\vec{A} \\
&= -\frac{1}{3}\int (\vec{\nabla}\cdot\vec{r}) p dV \\
&= -pV \qquad .
\end{aligned}$$

Aus der Zusammenfassung dieser Terme erhält man als Bestimmungsgleichung für den instantanen Druck $\mathcal{P}$

$$\mathcal{P}V = NkT + \langle W\rangle.$$

8.8.2 Koordinatenreskalierung

Man kann ein Gleichbleiben des Druckes dadurch erreichen, daß man in jedem Zeitschritt die Größe des Systems neu skaliert. Das bedeutet, daß man das Systemvolumen bei zu hohem Druck vergrößern und bei zu niedrigem Druck verkleinern muß,

$$\vec{r} \longrightarrow \beta\vec{r},$$

mit

$$\beta^3 = 1 - \alpha_T \frac{\Delta t}{t_p}(P - \mathcal{P}).$$

α_T ist die isotherme Kompressibilität.

8.8.3 Hoover

Die Gleichungen für das Hoover Verfahren bei konstantem Druck und konstanter Temperatur lauten:

$$\begin{aligned}
\dot{\vec{s}} &= \frac{\dot{\vec{r}}}{V^{1/3}} = \frac{\vec{p}}{mV^{1/3}} \\
\dot{\vec{p}} &= \vec{f} - (\chi + \xi)\vec{p} \\
\dot{\xi} &= \frac{fk}{Q}(\mathcal{T} - T) \\
\chi &= \frac{\dot{V}}{3V} \\
\dot{\chi} &= \frac{(\mathcal{P} - P)V}{t_p^2 kT}
\end{aligned}$$

Q und t_p sind Zeitkonstanten der "Regelung", die meist empirisch bestimmt werden [15].

8.9 Ereignisgesteuerte Molekulardynamik

8.9.1 Einführung

Nun betrachten wir starre Körper, die ein endliches Volumen haben und somit theoretisch am besten durch ein hard-core Potential beschrieben wären, was ja für die bisherigen MD Algorithmen eine große Schwierigkeit war, wegen der dabei auftretenden unendlich großen Kräfte. In der von Alder und Wainwright (1957) eingeführten "event-driven", d.h. ereignisgesteuerten Molekulardynamik betrachtet man die Kollisionen zwischen Teilchen als punktförmig und momentan, so daß es genügt, die Augenblicke zu bestimmen, zu denen eine Kollision stattfindet, d.h. zwei Teilchen sich berühren. Zwischen den Kollisionen fliegen die Teilchen ballistisch. Es finden also nur noch Berechnungen zum Zeitpunkt einer Teilchenkollision statt und nicht mehr alle Δt. Man macht somit folgende Idealisierungen:

- Eine Kollision hat keine endliche Dauer.
- Es werden keine Kräfte auf die Teilchen bestimmt.
- Kollisionen von drei oder mehr Teilchen werden nicht berücksichtigt.

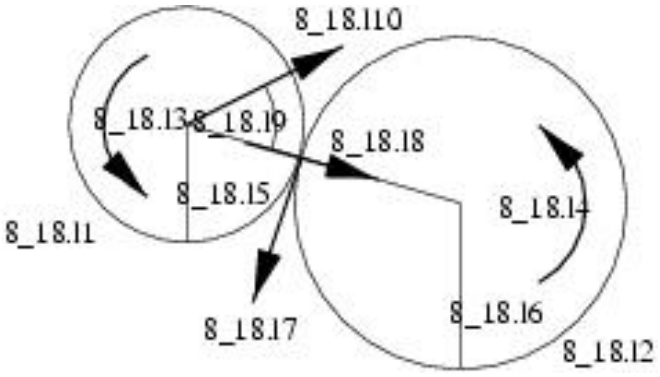


Abb. 8.11 *Kollision zweier Scheiben*

Der Einfachheit halber betrachten wir Billiardkugeln in 2 Dimensionen, d.h. Scheiben. Der Winkel θ zwischen der Relativgeschwindigkeit $\vec{v}_{ij} = \vec{v}_i - \vec{v}_j$ und der Verbindung der Schwerpunkte $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$ heißt Stoßwinkel.
Der Algorithmus besteht aus zwei Schritten: Der Berechnung der Zeit t_c zur nächsten im System stattfindenden Kollision und der Ausführung der Kollision. Um t_c zu bestimmen, muß man für jedes Paar (i, j) von Teilchen die Zeit t_{ij} bestimmen, zu der die Teilchen zusammenstoßen werden, wenn sie ballistisch fliegen. Die Kollision findet statt, wenn

$$\begin{aligned} |\vec{r}_{ij}(t_{ij})| &= R_i + R_j \\ \Longrightarrow |\vec{r}_{ij}(0) + \vec{v}_{ij} t_{ij}| &= R_i + R_j \end{aligned}$$

gilt, was zur quadratischen Gleichung

$$v_{ij}^2 t_{ij}^2 + 2(\vec{r}_{ij}\vec{v}_{ij})t_{ij} + r_{ij}^2 - (R_i + R_j)^2 = 0$$

führt. Deren Lösung kann man hinschreiben. Den Zeitpunkt der nächsten Kollision erhält man dann aus

$$t_c = \min_{ij} t_{ij}$$

Diese Operation ist auf dem Rechner sehr zeitraubend, da sie wie N^2 anwächst. Zudem ist eine globale Minimierung nicht parallelisierbar oder vektorisierbar. Gewisse Tricks (Lubachevsky, 92) erlauben es jedoch, den Algorithmus auf $Nlog(N)$ zu verschnellern. Nachdem t_c bekannt ist, bewegt man alle Teilchen gemäß

$$\vec{r}_i' = \vec{r}_i + \vec{v}_i t_c, \qquad \phi_i' = \phi_i + \omega_i t_c.$$

Nun tritt zum ersten Mal eine Kollision auf und es ist auch bekannt welches Teilchenpaar zusammenstößt.
Die ereignisgesteuerte Molekulardynamik ist am besten geeignet, wenn t_c groß ist im Vergleich zum entsprechenden Δt einer P-K oder Verlet Rechnung. Sehr langsam wird diese Methode, wenn die Teilchendichte zu hoch ist. Im dem Fall daß sich inelastische Teilchen permanent berühren, divergiert die Rechenzeit zu einem bestimmten Zeitpunkt sogar ("finite-time singularity").

8.9.2 Kollision mit perfektem Schlupf

Vorerst betrachten wir den Fall perfekten Schlupfes am Kontaktpunkt. Das bedeutet, daß zwischen den Teilchen keine Reibung besteht und somit auch kein Impulsübertrag in Tangentialrichtung stattfindet. Die Kollision wird durch die Geschwindigkeiten $\vec{v}_i^a$ vor der Kollision definiert. Man berechnet die Geschwindigkeiten $\vec{v}_i^e$ nach der Kollision aus der Impuls- und Energieerhaltung, sowie der Tatsache, daß der Impulsübertrag senkrecht auf der Kontaktfläche steht. Der Impulserhaltungssatz liefert

$$\vec{p}_i^a + \vec{p}_j^a = \vec{p}_i^e + \vec{p}_j^e.$$

Die Impulsänderung eines Teilchens ist somit der des anderen entgegengesetzt

$$\vec{p}_i^e = \vec{p}_i^a + \vec{\Delta p}, \qquad \vec{p}_j^e = \vec{p}_j^a - \vec{\Delta p}$$

Aus der Energierhaltung erhält man die Beziehung

$$m_i v_i^{a2} + m_j v_j^{a2} = m_i v_i^{e2} + m_j v_j^{e2}$$

Setzt man die Gleichung darüber hierin ein, ergibt sich

$$2(\vec{v}_i^a - \vec{v}_j^a)\vec{\Delta p} + \frac{\Delta p^2}{m_i} + \frac{\Delta p^2}{m_j} = 0,$$

und mit der Definition der effektiven Masse

$$m_{eff} = \frac{m_i m_j}{m_i + m_j}$$

schließlich

$$\vec{\Delta p}_n = -2m_{eff}[(\vec{v}_{ij} - \vec{v}_j^a)\vec{n}]\vec{n},$$

wobei schon die Projektion auf die Verbindung $\vec{n} = \frac{\vec{r}_{ij}}{r_{ij}}$ gebildet wurde. Die Geschwindigkeit nach dem Stoß berechnet sich schließlich nach

$$\vec{v}_i^e = \vec{v}_i^a + \frac{\vec{\Delta p}_n}{m_i}.$$

8.9.3 Kollision ohne Schlupf

Betrachtet man nun auch Drehungen der Teilchen, so ist die Geschwindigkeit am Kontaktpunkt

$$\vec{v}_{rel} = \vec{v}_j^a - \vec{v}_i^a + (R_i\omega_i + R_j\omega_j)\vec{t}$$

(mit $\vec{t}\cdot\vec{n} = 0$ und $|\vec{t}| = 1$)
die man in Normal- und Tangentialkomponente

$$\begin{aligned} v_n &= (\vec{v}_j - \vec{v}_i)\vec{n} \\ v_t &= (\vec{v}_j - \vec{v}_i)\vec{t} + R_i\omega_i + R_j\omega_j \end{aligned}$$

zerlegen kann. Bislang war $v_t = 0$ und $\vec{\Delta p} = \Delta p_n$. Nun hat $\vec{\Delta p}$ auch eine Tangentialkomponente. Die Tangentialkomponente des Impulsübertrags sorgt für eine Änderung des Drehimpulses

$$\begin{aligned} I_i(\vec{\omega}_i^e - \vec{\omega}_i^a) &= -\vec{\Delta p} \times \vec{n}R_i \\ \Longrightarrow \vec{\omega}_i^e &= \vec{\omega}_i^a - \frac{\vec{\Delta p} \times \vec{n}R_i}{I_i} \end{aligned}$$

Mit der letzten Formel berechnet man die neue Winkelgeschwindigkeit des Teilchens nach dem Stoß. Die dazu noch fehlende Größe $\vec{\Delta p}$ ergibt sich aus dem Energieerhaltungssatz

$$E = \frac{1}{2}\sum_i m_i v_i^2 + \frac{1}{2}\sum_i I_i\omega_i^2 = \text{const}$$

und der Erhaltung des Drehimpulses

$$J = \sum_i m_i(\vec{r}_i \times \vec{v}_i) = \sum_i I_i\vec{\omega}_i$$

zu

$$\vec{\Delta p} = -2m_{eff}[(\vec{v}_{ij}\vec{n})\vec{n} + \frac{I}{I + mR^2}(\vec{v}_{ij}\vec{t})\vec{t}].$$

Für gleich große Kugeln ergibt sich damit $\Delta p_s = \frac{2}{7}\Delta p_n$.

8.9.4 Inelastische Teilchen

In der Realität sind die Kollisionen zwischen Billiardkugeln inelastisch, da ein Teil der kinetischen Energie in Schwingungen (Schall), Wärme und eventuell kleine plastische Deformation der Oberfläche umgewandelt wird. Man mißt diese Dissipation in Normalenrichtung durch den materialabhängigen normalen Restitutionskoeffizienten

$$e_n = -\frac{v_n^e}{v_n^a} = (\frac{h^e}{h^a})^{\frac{1}{2}},$$

welcher bestimmt wird, indem man eine Kugel von einer Höhe h^a auf eine Platte gleichen Materials fallen läßt und die Höhe h^e mißt, welche die Kugel nach dem Aufprall wieder erreicht. $e_n = 1$ im elastischen Fall und liegt sonst zwischen 0 und 1.

	e_n
Stahl	0.93
Aluminium	0.8
Plastik	0.6

Bei der Kollision von zwei Teilchen gilt also

$$v_j^e - v_i^e = -e(v_j^a - v_i^a).$$

Die analoge Herleitung wie oben bei Kollisionen mit perfektem Schlupf liefert dann für den Impulsübertrag

$$\Delta p = -m_{eff}(e+1)(v_j^a - v_i^a).$$

Ohne Schlupf tritt eine weitere Dissipation durch eventuelle tangentiale Reibung auf, welche sich durch das Coulombsche Reibungsgesetz

$$\begin{array}{ll} |f_s| \leq \mu_h |f_n| & \text{falls} \quad v_s = 0 \\ |f_s| \leq \mu_g |f_n| & \text{falls} \quad v_s \neq 0 \end{array}$$

für die der Bewegung entgegengesetzten Reibungskraft f_s beschreiben läßt. Dabei ist μ_h der Haftreibungskoeffizient, μ_g der Gleitreibungskoeffizient und v_s die relative Tangentialgeschwindigkeit. Experimentell ist $\mu_h > \mu_g \approx 0.1 - 0.5$.

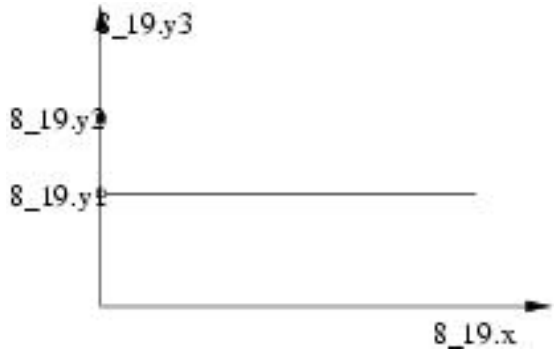


Abb. 8.12 *Verlauf des Reibungskoeffizienten in Abhängigkeit von der Tangentialgeschwindigkeit*

Bei der Simulation treten nun zwei Probleme auf:

1. Der Verlauf des Reibungskoeffizienten besitzt eine Singularität bei $v_s = 0$.
2. Wann erfolgt der Übergang von der Haft- zur Gleitreibung?

Für den tangentialen Impulsübertrag bei der Kollision von Kugeln ohne Schlupf erhält man

$$\begin{array}{ll} \Delta p_s = \frac{2}{7} m_{eff}(e_s - 1)v_s^a & \text{für} \quad v_s \ll 1 \\ \Delta p_s = \mu_g m_{eff}(1 + e_n)v_n^a & \text{für} \quad v_s \gg 1. \end{array}$$

e_s ist der tangentiale Restitutionskoeffizient, für den im Allgemeinen $e_s \approx \frac{1}{2} e_n$ gilt.

Zum Abschluß sei noch die Implementierung inelastischer Teilchen in eine "normale" Molekulardynamik besprochen. Man betrachtet eine elastische Abstoßungskraft und zusätzlich eine viskose Reibung

$$f_n = -kr - \gamma_n \dot{r} = m_{eff}\ddot{r},$$

mit einer Dämpfungskonstanten γ_n, wobei $r = R_i + R_j - |\vec{r}_i - \vec{r}_j|$ ist. Als Lösung dieser Gleichung erhält man

$$r(t) = \frac{v_n^a}{\omega} \sin(\omega t) e^{-\delta t},$$

mit

$$\omega = \sqrt{\omega_0^2 - \delta^2}, \qquad \omega_0^2 = \frac{k}{m_{eff}}, \qquad \delta = \frac{\gamma_n}{2m_{eff}}.$$

Die Kollisionszeit T ist ca. eine halbe Periode $\frac{\pi}{\omega}$, so daß für den normalen Restitutionskoeffizienten gilt

$$e_n = \frac{\dot{r}(t+T)}{\dot{r}(t)} = \exp(-\delta T) = \exp\left(-\frac{\gamma_n \pi}{\sqrt{4m_{eff}k - \gamma_n^2}}\right)$$

Man kann also im Falle linearer Abstoßungskräfte durch die Einführung einer viskosen Reibungskraft mit konstantem Reibungskoeffizienten γ_n einen konstanten Restitutionskoeffizienten simulieren und die entsprechende Dämpfungskonstante berechnen.
Die Tangentialkräfte, die die Coulombschen Reibung beschreiben, wurden von Cundall und Strack (1979) folgendermaßen in die Molekulardynamik implementiert:

$$\begin{aligned} f_s &= -\min(\gamma_s v_s, \mu_g f_n)\,\mathrm{sign}(v_s) \\ f_s &= -\min(|k_s \xi|, \mu_g f_n)\,\mathrm{sign}(v_s) \end{aligned}$$

wobei der erste Fall (dynamisch) einen tangentiellen Restitutionskoeffizienten durch die Dämpfungskonstante γ_s beschreibt, während im zweiten Fall (statisch) eine Feder

der Federkonstante k_s zwischen den Kontaktflächen angebracht wird, die dann eine Länge ξ gestreckt wird. Falls

$$|k_s \xi| > \mu_h f_n$$

gilt, reißt die Feder. Diese Methode führt allerdings zu unphysikalischen Oszillationen im Fall haftender Teilchen.

Kapitel 9

Lösung partieller Differentialgleichungen

9.1 Einführung

Man kann Differentialgleichungen nach der Art der vorgegebenen Werte und nach ihrer Form klassifizieren.

	Randwertprobleme	Anfangswertprobleme
skalare Gleichungen	$\Delta\Phi = \rho(\vec{x})$ Poissongleichung	$\frac{\partial^2\Phi}{\partial^2 t} = c^2\Delta\Phi$ Wellengleichung
	$\Delta\Phi = 0$ Laplacegleichung	$\frac{\partial\Phi}{\partial t} = \kappa\Delta\Phi$ Diffusionsgleichung
vektorielle Gleichungen	$\vec{\nabla}(\vec{\nabla}\vec{u}(\vec{x})) + (1-2\nu)\Delta\vec{u}(\vec{x}) = 0$ Lamégleichung	$\frac{\partial\vec{v}}{\partial t} + (\vec{v}\vec{\nabla})\vec{v} = -\frac{1}{\rho}\vec{\nabla}p + \nu\Delta\vec{v}$ Navier-Stokes-Gleichung

Die Poissongleichung, die Laplacegleichung und die Lamégleichung bezeichnet man als *elliptische* Gleichungen. Elliptische Gleichungen besitzen immer eine eindeutige Lösung (abgesehen von gewissen gemischten Cauchy-Randwertproblemen die keine Lösung besitzen). Die Wellengleichung ist ein *hyperbolische* Gleichung. Die Diffusionsgleichung ist eine *parabolische* Gleichung. Beide Arten von Differentialgleichungen besitzen nicht immer Lösungen.
Für Randwertprobleme unterscheidet man zwei Arten von Randbedingungen:

1. *Dirichletsche* Bedingungen, bei denen man die Werte Φ auf dem Rand vorgibt, und

2. *von Neumannsche* Randbedingungen, bei denen der Wert der ersten Ableitung auf dem Rand vorgegeben wird.

Das größte Problem bei der numerischen Lösung von Randwertaufgaben stellt das Erreichen einer ausreichend schnellen Konvergenz zur Lösung dar, während man bei Anfangswertaufgaben mit Stabilitätsproblemen zu kämpfen hat.

9.2 Exakte Lösung der Poissongleichung

In einer Dimension diskretisiert man die Poissongleichung folgendermaßen:

$$\begin{aligned} \frac{1}{\delta}\left(\frac{\Phi_{i+1}-\Phi_i}{\delta}-\frac{\Phi_i-\Phi_{i-1}}{\delta}\right) &= \rho_i \\ \Leftrightarrow \qquad \Phi_{i+1}+\Phi_{i-1}-2\Phi_i &= \delta^2\rho_i \qquad i=1\ldots N \end{aligned}$$

Man erhält ein System von N gekoppelten Gleichungen. Bei Dirichletsche Bedingungen gibt man

$$\Phi_0 = c_0, \qquad \Phi_{N+1} = c_1$$

vor und bei von Neumannsche Bedingungen

$$\Phi_0 - \Phi_1 = c_0, \qquad \Phi_N - \Phi_{N+1} = c_1.$$

In zwei Dimensionen betrachte man die Werte von Φ auf einem Quadratgitter. Die diskretisierte Laplacegleichung lautet dann

$$\Phi_{i+1\,j}+\Phi_{i-1\,j}+\Phi_{i\,j+1}+\Phi_{i\,j-1}-4\Phi_{ij}=\delta^2\rho_{ij} \quad \text{mit} \quad i=1,\ldots,L \quad j=1,\ldots,J.$$

Numerisch ist die Behandlung von eindimensionalen Feldern effektiver. Man ersetzt daher die Indizes i, j durch einen einzigen Index $k = i+(j-1)L$ und kann dann schreiben

$$\Phi_{k+L}+\Phi_{k-L}+\Phi_{k+1}+\Phi_{k-1}-4\Phi_k=\delta^2\rho_k.$$

Damit erhält man auch im zweidimensionalen Fall ein System von $N = LJ$ gekoppelten Gleichungen. Dieses System läßt sich nun mit Hilfe einer $N \times N$ Matrix $\mathbf{A}$ als

$$\mathbf{A}\vec{\Phi} = \vec{b}$$

schreiben. In einer Dimension hat das Gleichungssystem mit Dirichletschen Randbedingungen für $N = 4$ das Aussehen

$$\begin{pmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2 \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \end{pmatrix} = - \begin{pmatrix} c_0 \\ 0 \\ 0 \\ c_1 \end{pmatrix}.$$

Im zweidimensionalen Fall erhält man für ein 3×3 Quadratgitter bei Dirichletschen Randbedingungen, Φ_0 auf dem Rand

$$\begin{pmatrix} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{pmatrix} \vec{\Phi} = - \begin{pmatrix} 2 \\ 1 \\ 2 \\ 1 \\ 0 \\ 1 \\ 2 \\ 1 \\ 2 \end{pmatrix} \Phi_0$$

Jede Zeile hat hier maximal 5 Matrixelemente. Das bedeutet, daß im Fall großer Systeme die Matrix nur dünn besetzt ist, da die Dichte der nicht verschwindenden Matrixelemente wie $5/N$ abfällt.
Die Lösung obiger Matrixgleichung ist

$$\vec{\Phi}^* = \mathbf{A}^{-1}\vec{b}.$$

Um diese Lösung zu erhalten steht man also vor dem Problem, die Matrix **A** invertieren zu müssen. Mit dem Gaußschen Eliminationsvefahren kann man dieses Problem exakt lösen. Ziel dabei ist es, das Gleichungssystem

$$\begin{pmatrix} a_{1,1}\Phi_1 & \dots & a_{1N}\Phi_N \\ \vdots & & \vdots \\ a_{N1}\Phi_1 & \dots & a_{N,N}\Phi_N \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_N \end{pmatrix}$$

durch geeignete Addition von Vielfachen der Zeilen so zu vereinfachen, daß alle a_{ij} mit $i > j$ Null werden. Falls $a_{kk} = 0$ ist, wird Zeile k mit Zeile $l, l > k$ vertauscht. Falls das "Pivotelement" $a_{kk} \neq 0$ ist, wird folgendermaßen fortgefahren:

$$\begin{aligned} q_i &= -\frac{a_{ik}}{a_{kk}} \\ a_{ij} &= a_{ij} + q_i a_{kj}, \forall i, j > k \\ b_i &= b_i + q_i b_i \end{aligned}$$

Damit kann die Matrix A auf obere Dreiecksgestalt

$$\widetilde{A} = \begin{pmatrix} & \\ 0 & \\ & \end{pmatrix}$$

gebracht werden. Damit ist

$$\Phi_N = \frac{b_N}{a_{NN}}$$

und durch Rückwärtseinsetzen erhält man

$$\Phi_i = \frac{1}{\tilde{a}_{ii}} \left(\tilde{b}_i - \sum_{j=i+1}^{N} \tilde{a}_{ij} \Phi_j \right)$$

Der Rechenaufwand dafür ist ca. $0.25N^3$ Multiplikationen, Additionen und N^2 Divisionen, was jeweils ca. 1μsec CPU Zeit kostet. Für ein 100×100 Gitter hat man $N = 10^4$ was also eine CPU Zeit von ca. 12 Tagen ergibt. Die Grenze des Zumutbaren ist somit schon mit kleinen Matrizen schnell erreicht. Es gibt spezielle Programme, die Matrixoperationen schneller ausführen, wie das "Yale sparse matrix package". Eine Matrixinversion wächst hier nur mit $N^{5/2}$ an. Damit kann man sinnvoll Matrizen bis $N = 300^2$ behandeln. In jedem Fall sind diese Systeme für moderne Bedürfnisse jedoch zu klein, so daß man gezwungen ist, auf eine exakte Lösung zu verzichten, und numerische, approximative Methoden zu benutzen, die im Folgenden beschrieben werden.
Zum Abschluß sei jedoch noch darauf hingewiesen, daß die Matrix A "wohl-konditioniert" sein muß, um numerisch stabil zu sein. Was das heißt, wird an folgendem Gegenbeispiel erläutert. Gegeben sei das Gleichungssystem

$$\begin{pmatrix} 2.0 & 6.0 \\ 2.0 & 6.00001 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 8.0 \\ 8.00001 \end{pmatrix},$$

welches als Lösung $x = 1.0$ und $y = 1.0$ besitzt. Erlaubt man jedoch einen Rundungfehler der Form

$$\begin{pmatrix} 2.0 & 6.0 \\ 2.0 & 5.99999 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 8.0 \\ 8.00002 \end{pmatrix},$$

erhält man mit $x = 10.0$ und $y = -2.0$ eine Lösung, die weit von der eigentlichen Lösung abweicht.

9.3 Relaxationsmethoden

Eine anschauliche iterative Technik zur Annäherung an die Lösung der Laplacegleichung ist das Jacobiverfahren. Man führt jetzt eine Zeit t ein. Zur Zeit $t = 0$ startet man mit sinnvollen Anfangswerten $\Phi_{ij}(0)$ und bestimmt anschließend zu jedem Zeitschritt den Wert an jedem Gitterpunkt so, daß er die Gleichung lokal löst.

$$\Phi_{ij}(t+1) = \frac{1}{4} \left(\Phi_{i+1\,j}(t) + \Phi_{i-1\,j}(t) + \Phi_{i\,j+1}(t) + \Phi_{i\,j-1}(t) \right) - b_{i\,j}.$$

Da eine elliptische Gleichung eine eindeutige Lösung besitzt, konvergiert dieses Verfahren immer zu dieser Lösung. Zerlegt man die Matrix A in

$$\mathbf{A} = \mathbf{D} + \mathbf{U} + \mathbf{O},$$

wobei **D** nur die Diagonalelemente, **U** nur die Elemente unter der Diagonalen und **O** nur die Elemente über der Diagonalen enthält, so kann man das Verfahren allgemein durch

$$\vec{\Phi}(t+1) = \mathbf{D}^{-1}\left(\vec{b} - (\mathbf{O}+\mathbf{U})\,\vec{\Phi}(t)\right)$$

ausdrücken. Wie gewünscht ist der Fixpunkt $\vec{\Phi}^{\star}$ dieser Gleichung, genau die Lösung der Laplacegleichung.

$$\begin{aligned} & & \vec{\Phi}^{\star} &= \mathbf{D}^{-1}\left(\vec{b} - (\mathbf{O}+\mathbf{U})\,\vec{\Phi}^{\star}\right) \\ \Leftrightarrow & & \mathbf{D}\vec{\Phi}^{\star} &= \vec{b} - (\mathbf{O}+\mathbf{U})\,\vec{\Phi}^{\star} \\ \Leftrightarrow & & (\mathbf{D}+\mathbf{O}+\mathbf{U})\,\vec{\Phi}^{\star} &= \vec{b} \qquad \text{q.e.d.} \end{aligned}$$

Dieses Verfahren erreicht die exakte Lösung theoretisch nur nach unendlich vielen Iterationen. Man kann es jedoch abbrechen, wenn man mit der erreichten Genauigkeit zufrieden ist. Als Maß für die Genauigkeit kann man folgende Größe einführen.

$$\Delta' = \frac{\|\vec{\Phi}(t+1) - \vec{\Phi}(t)\|}{\|\vec{\Phi}(t)\|} \leq \varepsilon$$

Ist die Ungleichung erfüllt,bricht man die Rechnung ab. Dabei heißt ε die Toleranz und $\|\ldots\|$ sei die Vektornorm. Das Jacobiverfahren ist leicht zu parallelisieren, doch es konvergiert sehr langsam. Dies liegt daran, daß Störungen stets hin und zurückgeschoben werden, während sie diffusiv zerfallen. Eine etwas andere Methode, das Gauß -Seidel Verfahren (GSV), verbessert dies, indem die Korrekturwelle nur noch nach unten und rechts geschoben wird, also das Oszillieren unterdrückt wird. Die allgemeine Iterationsgleichung des GSV lautet

$$\vec{\Phi}(t+1) = (\mathbf{D}+\mathbf{O})^{-1}\left(\vec{b} - \mathbf{U}\vec{\Phi}(t)\right).$$

Durch das Einsetzen des Fixpunktes $\vec{\Phi}^{\star}$ zeigt man, daß auch dieses Verfahren gegen die Lösung der Laplacegleichung konvergiert, d.h.

$$\Phi_i(t+1) = -\frac{1}{a_{ii}}\left(\sum_{j=i+1}^{N} a_{ij}\Phi_j(t) + \sum_{j=1}^{i-1} a_{ij}\Phi_j(t+1) - b_i\right)$$

Mit Hilfe der Beziehung

$$\begin{aligned} \mathbf{D}^{-1} - \mathbf{A}^{-1} &= \mathbf{D}^{-1}\mathbf{D}(\mathbf{D}^{-1} - \mathbf{A}^{-1})\mathbf{A}\mathbf{A}^{-1} \\ &= \mathbf{D}^{-1}(\mathbf{A}-\mathbf{D})\mathbf{A}^{-1} \\ &= \mathbf{D}^{-1}(\mathbf{O}+\mathbf{U})\mathbf{A}^{-1} \end{aligned}$$

kann man die Zeitentwicklung des Fehlers des Jacobi-Verfahrens bestimmen.

$$\begin{aligned}\vec{\varepsilon}(t+1) &= \underbrace{\mathbf{A}^{-1}\vec{b}}_{\text{exakte Lösung}} - \underbrace{\mathbf{D}^{-1}(\vec{b}-(\mathbf{O}+\mathbf{U})\vec{\Phi}(t))}_{\text{genäherte Lösung}} \\ &= -\mathbf{D}^{-1}(\mathbf{O}+\mathbf{U})\underbrace{(\mathbf{A}^{-1}\vec{b}-\vec{\Phi}(t))}_{\vec{\varepsilon}(t)} \\ &= -\mathbf{D}^{-1}(\mathbf{O}+\mathbf{U})\vec{\varepsilon}(t)\end{aligned}$$

Analog bestimmt man unter Ausnutzung von

$$\begin{aligned}(\mathbf{D}+\mathbf{O})^{-1}-\mathbf{A}^{-1} &= (\mathbf{D}+\mathbf{O})^{-1}(\mathbf{D}+\mathbf{O})[(\mathbf{D}+\mathbf{O})^{-1}-\mathbf{A}^{-1}]\mathbf{A}\mathbf{A}^{-1} \\ &= (\mathbf{D}+\mathbf{O})^{-1}\mathbf{U}\mathbf{A}^{-1}\end{aligned}$$

die Zeitentwicklung des Fehlers beim Gauß-Seidel-Verfahren.

$$\begin{aligned}\vec{\varepsilon}(t+1) &= \mathbf{A}^{-1}\vec{b}-(\mathbf{D}+\mathbf{O})^{-1}(\vec{b}-\mathbf{U}\vec{\Phi}(t)) \\ &= -(\mathbf{D}+\mathbf{O})^{-1}\mathbf{U}(\mathbf{A}^{-1}\vec{b}-\vec{\Phi}(t)) \\ &= -\underbrace{(\mathbf{D}+\mathbf{O})^{-1}\mathbf{U}}\vec{\varepsilon}(t)\end{aligned}$$

Der größte Eigenwert der unterklammerten Matrix sei λ mit $0<\lambda<1$. Man kann dann die Näherungslösung zum Zeitpunkt t selber wieder annähern durch

$$\vec{\Phi}(t)=\vec{\Phi}^{\star}+\vec{c}\lambda^{t-1}.$$

$\vec{\Phi}^{\star}$ ist die exakte Lösung. Da $\vec{\Phi}(t)$, wie oben bewiesen, gegen die exakte Lösung konvergiert, muß für λ gelten, daß $0<\lambda<1$. Man kann nun den relativen Fehler umformen wie

$$\Delta_t'=\frac{\|\vec{\Phi}(t+1)-\vec{\Phi}(t)\|}{\|\vec{\Phi}(t)\|}\approx\frac{\|\vec{c}(\lambda^t-\lambda^{t-1})\|}{\|\vec{\Phi}(t)\|}=\frac{\|\vec{c}\|}{\|\vec{\Phi}(t)\|}|\lambda^{t-1}(\lambda-1)|.$$

Der wirkliche Fehler ist allerdings

$$\Delta_t=\frac{\|\vec{\Phi}^{\star}-\vec{\Phi}(t)\|}{\|\vec{\Phi}(t)\|}\approx\frac{\|\vec{c}\|}{\|\vec{\Phi}(t)\|}\lambda^{t-1},$$

so daß

$$\Delta_t'=(1-\lambda)\Delta_t.$$

Wegen $0<\lambda<1$ unterschätzt der relative Fehler also den wirklichen Fehler. Das wird gefährlich im Falle kritischer Relaxation, d.h. wenn $\lambda\approx 1$. Deshalb ist es im Allgemeinfall sicherer λ durch

$$\frac{\|\vec{\Phi}(t+1)-\vec{\Phi}(t)\|}{\|\vec{\Phi}(t)-\vec{\Phi}(t-1)\|}\approx\frac{\lambda^{t+1}-\lambda^t}{\lambda^t-\lambda^{t-1}}=\lambda$$

zu bestimmen und dann den wirklichen Fehler

$$\Delta_t = \frac{\Delta'_t}{1-\lambda} = \frac{\|\vec{\Phi}(t) - \vec{\Phi}(t-1)\|^2}{\|\vec{\Phi}(t)\|(\|\vec{\Phi}(t) - \vec{\Phi}(t-1)\| - \|\vec{\Phi}(t+1) - \vec{\Phi}(t)\|)}$$

im Abbruchkriterium zu benutzen.
Eine weitere Verbesserung des Gauß -Seidel Verfahrens erreicht man durch Überrelaxation ("successive overrelaxation", SOR). Die Iteration lautet

$$\vec{\Phi}(t+1) = (\mathbf{D} + \omega\mathbf{O})^{-1}(\omega\vec{b} + ((1-\omega)\mathbf{D} - \omega\mathbf{U})\vec{\Phi}(t))$$

wobei ω der sogenannte Überrelaxationsparameter ist. Durch Einsetzen kann man zeigen, daß der Fixpunkt der Gleichung die gesuchte Lösung ist. Man kann meist ein $1 < \omega \leq 2$ empirisch bestimmen, für welches die Konvergenz schneller als für den Fall $\omega = 1$ (Gauß -Seidel) ist.

9.4 Anwendung von Relaxationsmethoden

Die Lösung der Laplacegleichung auf einem Gitter durch Relaxationsmethoden hat auch mehrere direkte Anwendungen in der physikalischen Modellierung. Im elektrostatischen Fall kann man sich vorstellen, daß die Verbindungen zwischen den Gitterplätzen des Gitters Drähte mit einem bestimmten Widerstand sind. Dann ist die diskretisierte Laplacegleichung gerade die Erhaltung des Kichhoffschen Gesetzes an jedem Gitterpunkt. Solche Netze von Widerständen sind auch von Interesse bei der Bestimmung der Permeabilität poröser Medien. Die Unordnung kann man durch zufällig gewählte Widerstände oder durch (zufällig) fehlende Verbindungen beschreiben. Besonders interessant ist die Anwendung der Relaxationsmethoden auf nichtlineare Widerstände

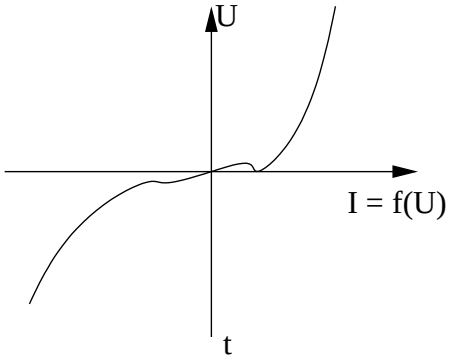


Abb. 9.1 *Kennlinie eines nichtlinearen Widerstandes*

was eine diskrete Gleichung der Form

$$f(\Phi_{i-1\,j} - \Phi_{i\,j}) + f(\Phi_{i+1\,j} - \Phi_{i\,j}) + f(\Phi_{i\,j-1} - \Phi_{i\,j}) + f(\Phi_{i\,j+1} - \Phi_{i\,j}) = 0$$

ergibt.
Eine andere direkte Anwendung von Relaxationsmethoden auf die diskrete Laplacegleichung ist die Diffusion. Man kann die zeitliche Entwicklung der Wahrscheinlichkeitswolke eines Zufallsweges, in dem das Teilchen an jedem Zeitschritt mit Wahrscheinlichkeit $\frac{1}{4}$ auf einen der 4 benachbarten Plätze des Quadratgitters springt, direkt durch Jacobirelaxation bestimmen (Ben-Avraham und Havlin, 1982)

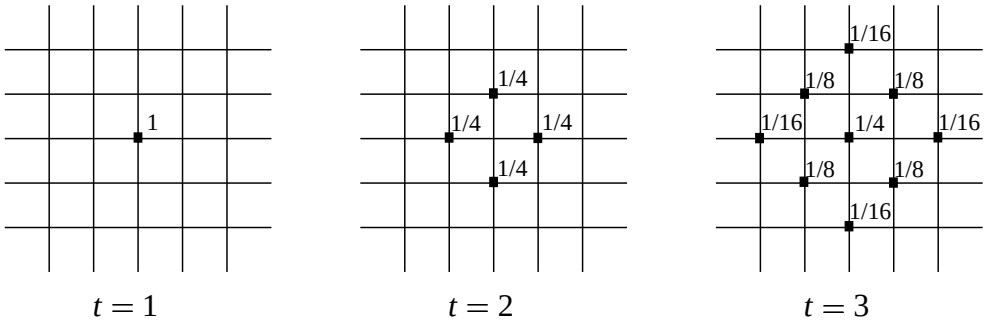


Abb. 9.2 *Zeitliche Entwicklung einer Wahrscheinlichkeitswolke*

Eine weitere Anwendung ist das Modell der "selbstorganisierten Kritizität" (SOC). Hier wird nur nach Jacobi relaxiert, wenn die Variable auf einem Gitterplatz einen Wert z überschreitet. Dabei können Kettenreaktionen entstehen. Man kann daher mit diesem Modell Lawinen und Erdbeben beschreiben (Bak, Tang und Wiesenfeld, 1988).

9.5 Gradientenmethoden

9.5.1 Minimalisierung des Fehlers

Wir definieren das Residuum der genäherten Lösung $\vec{\Phi}$ durch

$$\vec{r} = \vec{b} - \mathbf{A}\vec{\Phi}.$$

$\vec{r}$ ist mit dem Fehler von Φ über $\vec{\varepsilon} = \mathbf{A}^{-1}\vec{r}$ verknüpft und ist somit ein anderes Maß für den Fehler. Nun führt man das Funktional $\mathcal{F}$ durch

$$\mathcal{F} = \vec{r}^t \mathbf{A}^{-1}\vec{r}$$

ein. $\mathbf{A}$ sei positiv definit und symmetrisch. Dann gilt

$$\mathcal{F}(\vec{\Phi}) = \begin{cases} 0 & \text{falls } \vec{\Phi} = \vec{\Phi}^\star \\ > 0 & \text{sonst} \end{cases}$$

D.h., daß man durch die Minimalisierung dieses Funktionals die exakte Lösung erhalten sollte. Durch Einsetzen erhält man

$$\begin{aligned} \mathcal{F} &= (\vec{b} - \mathbf{A}\vec{\Phi})^t \mathbf{A}^{-1}(\vec{b} - \mathbf{A}\vec{\Phi}) \\ &= \vec{b}^t \mathbf{A}^{-1}\vec{b} + \vec{\Phi}^t \mathbf{A}\vec{\Phi} - 2\vec{b}\vec{\Phi} \end{aligned}$$

Die Minimalisierung wird schrittweise auf Linien

$$\vec{\Phi} = \vec{\Phi}_i + \alpha_i \vec{d}_i$$

durchgeführt, wobei $\vec{\Phi}_i$ die angenäherte Lösung des i-ten Schrittes und $\vec{d}_i$ der vorerst noch nicht festgelegte Richtungsvektor der i-ten Linie sei. Den Parameter α_i wollen wir durch Minimalisierung von $\mathcal{F}$ bestimmen. Dazu setzen wir die Liniengleichung in $\mathcal{F}$ ein.

$$\mathcal{F} = \alpha_i^2 \vec{d}_i^t \mathbf{A} \vec{d}_i + 2\vec{\alpha}_i d_i^t \mathbf{A} \vec{\Phi}_i + \vec{\Phi}_i^t \mathbf{A} \vec{\Phi}_i - 2\vec{b}^t \vec{\Phi}_i - 2\alpha_i \vec{b}^t \vec{d}_i + \vec{b}^t \mathbf{A}^{-1} \vec{b}$$

und benutzen die Gleichung

$$\frac{\partial \mathcal{F}}{\partial \alpha_i} = 2\vec{d}_i^t (\alpha_i \mathbf{A} \vec{d}_i - \vec{r}_i) = 0$$

$$\Rightarrow \qquad \alpha_i = \frac{\vec{d}_i^t \vec{r}_i}{\vec{d}_i^t \mathbf{A} \vec{d}_i}$$

Folgendes Diagramm, in dem zwei Komponenten von $\vec{\Phi}$ gegeneinender aufgetragen sind und $\mathcal{F}$ als Höhenlinie dargestellt wird, erläutert das Vorgehen

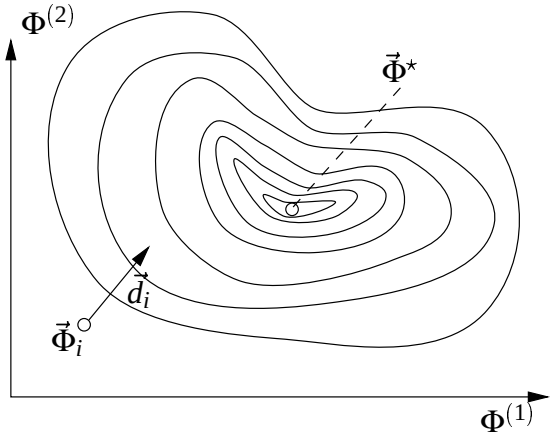


Abb. 9.3 *Höhenlinien von $\mathcal{F}$*

Je länger die Täler, d.h. je verschiedener die Eigenwerte von A, desto schlechter ist die Konvergenz der Gradientenmethoden. Falls man für $\vec{d}_i$ die orthogonale Basis nimmt, in der die Matrix A dargestellt ist, so erhält man durch obiges Schema gerade die Gaußelimination.

9.5.2 Methode des steilsten Abfalls (steepest descent)

Hier wählt man

$$\vec{d}_i = \vec{r}_i.$$

Nimmt man einen Anfangsvektor $\vec{\Phi}_1$, so implementiert man diesen Algorithmus also folgendermaßen

$$\begin{aligned}
\vec{r}_1 &= \vec{b} - \mathbf{A}\vec{\Phi}_1 \\
\vec{u}_i &= \mathbf{A}\vec{r}_i \\
\alpha_i &= \frac{\vec{r}_i^2}{\vec{r}_i^t \vec{u}_i} \\
\vec{\Phi}_{i+1} &= \vec{\Phi}_i + \alpha_i \vec{r}_i \\
\vec{r}_{i+1} &= \vec{r}_i - \alpha_i \vec{u}_i.
\end{aligned}$$

Danach fängt man wieder bei der 2.Zeile an. Der Aufwand der letzten drei Zeilen wächst $\propto N$. Der Aufwand der zweiten Zeile wächst $\propto N^2$. Man kann den Rechenaufwand für diese Zeile jedoch auf kN reduzieren, wenn die Matrix **A** dünn besetzt ist und nur k nichtverschwindende Matrixelemente pro Reihe hat. Die Matrix **A** braucht man in diesem Falle auch nicht abzuspeichern. Diese Tricks gelten auch für die Methode des nächsten Abschnitts.

9.5.3 Konjugierter Gradient (Hestenes und Stiefel, 1952)

Die Konvergenz der Gradientenmethode wird, wie bereits erwähnt, umso schlechter, je länger das Tal ist, in dem die Lösung liegt. Anschaulich gedeutet, besteht die Idee der Methode des konjugierten Gradienten nun darin, eine geeignete Metrik so zu finden, daß das Tal isotrop wird. Dazu wählt man die $\vec{d}_i$ zueinander konjugiert.

$$\vec{d}_i^t \mathbf{A} \vec{d}_j = \delta_{ij}$$

Für das Residuum $\vec{r}$, den Schrittweiteparameter α und die Lösung gilt weiterhin

$$\begin{aligned}
\vec{r}_i &= \vec{b} - \mathbf{A}\vec{\Phi}_i \\
\alpha_i &= \frac{\vec{r}_i \vec{d}_i}{\vec{d}_i^t \mathbf{A} \vec{d}_i} \\
\vec{\Phi}_{i+1} &= \vec{\Phi}_i + \alpha_i \vec{d}_i.
\end{aligned}$$

Damit kann man das Residuum auch schreiben als

$$\vec{r}_i = \vec{b} - \mathbf{A}(\vec{\Phi}_1 + \sum_{j=1}^{i-1} \alpha_i \vec{d}_i).$$

Mit der letzten Gleichung und der Bedingung, die man an die $\vec{d}_i$ gestellt hat, kann man nun zeigen, daß gilt

$$\vec{r}_i^t \mathbf{A} \vec{d}_j = \delta_{ij}.$$

Anschaulich bedeutet das aber, daß der Fehler im Schritt i keine Komponenten mehr in Richtung der bereits gemachten Schritte besitzt. D.h. nach N Schritten, wobei N die Dimension der Vektoren ist, konvergiert das Verfahren in die *exakte* Lösung.
Die $\vec{d}_i$ bestimmt man nun iterativ mit dem Schmidtschen Orthogonalisierungsverfahren.

$$\begin{aligned}\vec{d}_i &= \vec{r}_1 \\ \vec{d}_i &= \vec{r}_i - \sum_{j=1}^{i-1} \frac{\vec{d}_j^t \mathbf{A} \vec{r}_i}{\vec{d}_j^t \mathbf{A} \vec{d}_j^t} \vec{d}_j\end{aligned}$$

Zusammengefasst erhält man also folgende Vorgehensweise:

1. $\vec{\Phi}_1$ sei gegeben. Dann bestimmt man

$$\begin{aligned}\vec{r}_1 &= \vec{b} - \mathbf{A}\vec{\Phi}_1 \\ \vec{d}_1 &= \vec{r}_1.\end{aligned}$$

2. Anschließend durchläuft man in einer Schleife die folgenden Schritte:

$$\begin{aligned}c &= (\vec{d}_i^t \mathbf{A} \vec{d}_i)^{-1} \\ \alpha_i &= c\vec{d}_i \vec{r}_i \\ \vec{\Phi}_{i+1} &= \vec{\Phi}_i + \alpha_i \vec{d}_i \\ \vec{r}_{i+1} &= \vec{b} - \mathbf{A}\vec{\Phi}_{i+1} \\ \vec{d}_{i+1} &= \vec{r}_{i+1} - (c\vec{r}_{i+1}\mathbf{A}\vec{d}_i)\vec{d}_i\end{aligned}$$

3. Als Abbruchkriterium benutzt man

$$\vec{r}_i^t \vec{r}_i < \varepsilon$$

Ist die Matrix $\mathbf{A}$ schlecht konditioniert (also numerisch, d.h. mit endlicher Rechengenauigkeit nicht invertierbar), dann kann durch die Wahl einer Matrix $\tilde{\mathbf{A}}$ mit der Eigenschaft $\tilde{\mathbf{A}}^{-1}\mathbf{A} \approx \mathbf{1}$ das Verhalten des Verfahrens verbessern. Man hat dann das Gleichungssystem $(\tilde{\mathbf{A}}^{-1}\mathbf{A})\vec{\Phi} = \tilde{\mathbf{A}}^{-1}\vec{b}$ zu lösen.
Beispiel:
Gegeben sei $\mathbf{A} = \begin{pmatrix} \lambda_1 & \tau_2 \\ \tau_1 & \lambda_2 \end{pmatrix}$ mit τ_1, τ_2 klein und $\lambda_1 \gg \lambda_2$
Nun wählt man $\tilde{\mathbf{A}} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$.
Das neue Gleichungssystem hat dann die Form

$$\begin{pmatrix} 1 & \frac{\tau_2}{\lambda_1} \\ \frac{\tau_1}{\lambda_2} & 1 \end{pmatrix} \vec{\Phi} = \begin{pmatrix} \frac{b_1}{\lambda_1} \\ \frac{b_2}{\lambda_2} \end{pmatrix}$$

9.6 Mehrgitterverfahren (Brandt 1970)

Bei numerischen Verfahren zur Lösung von Gleichungen der Form

$$\mathbf{A}\vec{\Phi} = \vec{b},$$

gilt für die numerische Lösung näherungsweise

$$\mathbf{A}\vec{\Phi}_i \approx \vec{b}.$$

Bei der Herleitung von Mehrgitterverfahren folgt daraus die Gleichung für den Fehler $\vec{\varepsilon} = \mathbf{A}^{-1}\vec{b} - \vec{\Phi}_i$, und man definiert das Residuum $\vec{r} = \vec{b} - \mathbf{A}\vec{\Phi}_i$. Damit ist die Gleichung für den Fehler

$$\mathbf{A}\vec{\varepsilon} = \vec{r}.$$

Die Funktionsweise eines Mehrgitterlösers sei nun am Beispiel eines 2-Gitter-Verfahrens dargestellt. Ziel ist es, das Residuum durch das hin- und herwechseln zwischen feinem und groben Gitter zu reduzieren.

1. Man bestimmt zunächst $\vec{r}$ auf dem ursprünglichen Gitter.
2. Mit Hilfe eines Projektionsoperators $\mathcal{R}$, dem sogenannten *Restriktionsoperator*, auf den später näher eingegangen werden soll, projeziert man $\vec{r}$ auf ein gröberes Gitter.

   $$\hat{\vec{r}} = \mathcal{R}\vec{r}.$$

   $\hat{}$bezeichne hier und im Folgenden die auf das grobe Gitter abgebildeten Größen.
3. Auf dem groben Gitter löst man nun

   $$\hat{\mathbf{A}}\hat{\vec{\varepsilon}} = \hat{\vec{r}}.$$

4. Das Ergebnis bildet man durch eine *Erweiterung* wieder auf das feine Gitter ab.

   $$\vec{\varepsilon} = \mathcal{P}\hat{\vec{\varepsilon}}$$

5. Für die Näherungslösung im nächsten Schritt gilt dann

   $$\vec{\Phi}_{i+1} = \vec{\Phi} + \vec{\varepsilon}$$

Die bisher behandelten Relaxationsmethoden arbeiten sehr lokal. Das heißt, daß sie gut dazu geeignet sind, kurzwellige Störungen der Lösung herauszufiltern. Die Relaxation von langwelligen Störungen braucht dagegen sehr lang. Genau diesen Mangel behebt man durch den Übergang auf ein grobes Gitter. Auf dem groben Gitter werden langwellige Störungen sehr schnell herausgefiltert, wogegen nun jedoch die kurzwelligen Störungen nicht mehr aufgelöst werden. Ein komplettes Mehrgitterverfahren kombiniert daher beide Methoden.

1. Man startet mit einer ersten Approximation $\vec{\Phi}_1$ der Lösung und berechnet den Fehler. Noch auf dem feinsten Gitter unterzieht man den Fehler einer ersten "Vorglättung".

2. Nun geht man sukzessive auf gröbere Gitter, wobei man den Fehler auf jedem Gitter einige Iterationen glättet.

3. Auf dem gröbsten Gitter löst man das Gleichungssystem nun exakt z.B. durch Matrixinversion.

4. Beim anschließenden "Abstieg" auf das feinste Gitter führt man auf jeder Stufe nochmals ein Nachglätten durch.

Als Glätter verwendet man z.B. das Gauß-Seidel-Verfahren. Ungeeignet ist die Methode der Überrelaxation. Man bezeichnet diese Vorgehensweise mit einem Auf- und einem Abstieg als *V-Zyklus*. Daneben gibt es noch *W-Zyklen*, die entsprechend mehrmals auf- und absteigen.
Es bleiben noch die Operatoren der Erweiterung und der Restriktion zu besprechen. Ein mögliches Verfahren der Erweiterung, d.h. des Übergangs vom groben zum feinen Gitter, stellt die *bilineare Interpolation* von der Lösung auf dem gröberen Gitter $\hat{r}_{ij}$ zur Lösung auf dem feineren Gitter r_{ij} dar.

$$\mathcal{P}\hat{\vec{r}} \mapsto \begin{cases} r_{2i,2j} & = & \hat{r}_{i,j} \\ r_{2i+1,2j} & = & \frac{1}{2}\left(\hat{r}_{i,j}+\hat{r}_{i+1,j}\right) \\ r_{2i,2j+1} & = & \frac{1}{2}\left(\hat{r}_{i,j}+\hat{r}_{i,j+1}\right) \\ r_{2i+1,2j+1} & = & \frac{1}{4}\left(\hat{r}_{i,j}+\hat{r}_{i+1,j}+\hat{r}_{i,j+1}+\hat{r}_{i+1,j+1}\right) \end{cases}$$

Die Umkehrung der "Erweiterungen" (Interpolationen) auf das feinere Gitter sind die "Restriktionen" auf das gröbere Gitter. Die einfachste Möglichkeit der Restriktion besteht darin, den Wert der Gitterplätze weiterzugeben, welche im gröberen Gitter erhalten bleiben. Die übrigen Plätze werden vernachlässigt. Diese sogenannte *Injektion* ist jedoch schlecht, da Information verloren geht. Eine bessere Möglichkeit stellt die zu $\mathcal{P}$ adjunkte Wahl von $\mathcal{R}$ dar. $\mathcal{R}$ heißt dann adjunkt zu $\mathcal{P}$, wenn die Gleichung

$$\sum_{xy} \mathcal{P}\hat{r}(\hat{x},\hat{y})u(x.y) = h^2 \sum_{\hat{x}\hat{y}} \hat{r}(\hat{x},\hat{y})\mathcal{R}u(x,y)$$

erfüllt ist. h ist der Skalenfaktor. Aus $\hat{x} = hx$ sieht man, daß für ein grobes Gitter mit z.B. doppelt so breiten Maschen, $h = 2$ gewählt werden muß. Man erhält nun

$$\hat{r}_i = \frac{1}{4}r_i + \frac{1}{8}\sum_{j=nn(i)} r_j + \frac{1}{16}\sum_{j=nnn(i)} r_j.$$

Eine ausführlichere Beschreibung von Mehrgitterverfahren entnehme man der Literatur, z.B. [18].

9.7 Fourierbeschleunigung

Als Iterationsschritt bei der Jacobirelaxation hatte man

$$\begin{aligned}\vec{\Phi}_{i+1} &= \mathbf{D}^{-1}(\vec{b}-(\mathbf{O}+\mathbf{U})\vec{\Phi}_i)\\ &= \vec{\Phi}_i+\mathbf{D}^{-1}(\vec{b}-\mathbf{A}\vec{\Phi}_i)\\ &= \vec{\Phi}_i+\mathbf{D}^{-1}\vec{r}.\end{aligned}$$

Unter Einführung einer neuen Matrix, dem sogenannten Relaxationsparameter ε kann man die letzte Gleichung auch schreiben als

$$\vec{\Phi}_{i+1}=\vec{\Phi}_i+\varepsilon\vec{r}.$$

Im Falle der Jacobirelaxation wäre $\varepsilon=\mathbf{D}^{-1}$. Die exakte Lösung erhält man, wenn man $\varepsilon=\mathbf{A}^{-1}$ wählt. Von O'Slaughness und Procaccia (1980) stammt der Vorschlag,

$$\varepsilon_{ij}=r_{ij}^{2-d}$$

zu wählen aufgrund der Ähnlichkeit des Jacobiverfahrens mit der Diffusion. d ist die Dimension des Gitters, r ist der Abstand der Punkte im Raum. Nun führt man die Fouriertransformation

$$\mathcal{F}(\mathbf{A})=\sum_{lj}e^{i\vec{k}\vec{r}_{lj}}\mathbf{A}_{lj}$$

ein. Die Fouriertransformierte von ε ist dann

$$\varepsilon_k=\mathcal{F}((\varepsilon_{ij}))\propto k^{-2},$$

und der Iterationsschritt bekommt nun die Form

$$\vec{\Phi}_{i+1}=\vec{\Phi}_i+\mathcal{F}^{-1}(\varepsilon_k\mathcal{F}(\vec{b}-\mathbf{A}\vec{\Phi}_i)_k).$$

Die Methode findet vor allem auf unregelmäßigen Gittern Anwendung, wobei man die Konvolution

$$\mathcal{F}(\mathbf{A})\mathcal{F}(\mathbf{B})=\mathcal{F}(\mathbf{AB})$$

benutzt hat. Dies ist nützlich, falls das System fraktal ist: $\varepsilon_k\propto k^{dw}, dw>2$. Dadurch werden mit der Abhängigkeit k^{-2} langwellige Moden weggedämpft. Eine Kombination von Konjugierten Gradienten und Fourierbeschleunigung findet sich in G.G.Batrouni, A. Hansen, J. Stat. Phys. **52**, p. 747 (1988).

9.8 Navier-Stokes-Gleichung

Motivation

Bei den bisher behandelten Gleichungen, der Laplace- und Poisson-Gleichung, handelte es sich um skalare partielle Differentialgleichungen ohne Zeitabhängigkeit. Sie beschrieben z.B. das elektrische Potential oder die Konzentration eines Stoffes, wenn die Werte der Konzentration oder des Potentials am Rande des Gebietes Ω vorgegeben war, auf dem die Lösung gesucht wurde. Die Poisson-Gleichung enthielt des weiteren noch einen Quellterm (im Falle des elektrischen Potentials die entsprechende Ladungsverteilung), der ortsabhängig oder/und zeitabhängig sein kann.
Wir wollen jetzt Gleichungen betrachten, die Ableitungen nach der Zeit enthalten. Als Beispiel einer linearen Gleichung nennen wir hier die Diffusionsgleichung, die die zeitliche Entwicklung der Konzentration einer Stoffmenge c beschreibt, die sich durch Brownsche Bewegung getrieben langsam in alle Raumrichtungen ausbreitet,

$$\frac{\partial}{\partial t} c = D \underline{\nabla}^2 c, \tag{9.1}$$

wobei hier D die Diffusionkonstante bezeichnet. Die vollständige Spezifikation des Problem beinhaltet hier die Angabe der Funktionswerte $c(t)$ für $t = 0$ und die Werte auf dem Rand $\partial\Omega$ des Lösungsgebietes Ω für alle Zeitpunkte t. Wir werden im folgenden sehen, daß diese Gleichung ein Spezialfall der Formulierung einer Erhaltungsgleichung ist, hier für die gesamte Stoffmenge im Integrationsgebiet.
Weiterhin werden wir mit der Navier-Stokes-Gleichung einer Gleichung begegnen, die die Zeitentwicklung von Größen vektoriellen Charakters beschreibt. Bereits im letzten Jahrhundert erkannte man ihre zentrale Rolle für die Beschreibung von Flüssigkeiten mit innerer Reibung. Im folgenden werden wir zunächst eine allgemeine Herleitung angeben, bevor wir uns mit der numerischen Behandlung auseinandersetzen.

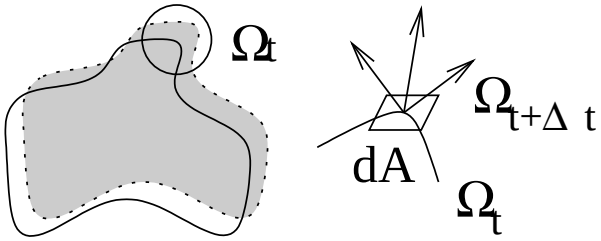
Das Reynoldsche Transporttheorem

Zur Herleitung der Navier-Stokes-Gleichung müssen wir uns zunächst mit der Frage beschäftigen, wie die Zeitabhängigkeit von Integralen ist, bei denen sowohl der Integrand als auch das Integrationsgebiet zeitabhängig sind. Diese Frage führt uns auf das Reynoldsche Transporttheorem. Die Bewegung des Integrationsgebietes soll durch ein "Geschwindigkeitsfeld" $\underline{v}(\underline{x}, t)$ beschrieben werden, das überall im Raum definiert ist. Dieses kann, muß aber nicht, mit der physikalischen Geschwindigkeit des Mediums zusamenfallen, das wir beschrieben wollen. Wir wollen die Zeitabhängigkeit des Integrationsgebietes Ω_t durch den Index t kennzeichnen. An jedem Punkt $\underline{x}$ des Randes $\partial\Omega_t$ bildet $\underline{v}(\underline{x}, t)$ die Flächennormale. Im folgenden werden wir der Einfachheit halber die Abhängigkeit von $\underline{v}$ nicht explizit ausschreiben. Neben dem Term, der wie üblich durch die Vertauschung von Ableitung und Integration entsteht, steht jetzt ein zweiter Beitrag, der die zeitliche Änderung des Integrationsgebietes berücksichtigt.

Für eine skalare Größe $f(\underline{x},t)$ lautet das Transporttheorem

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_t} f(\underline{x},t)\mathrm{d}\underline{x} = \int_{\Omega_t}\left\{\frac{\partial}{\partial t}f + \underline{\nabla}\cdot(f\underline{v})\right\}(\underline{x},t)\,\mathrm{d}\underline{x} \tag{9.2}$$

Dabei ist $\frac{\partial}{\partial t}f$ der Beitrag aufgrund der Änderung von f mit der Zeit t, und $\underline{\nabla}\cdot(f\underline{v})$ ist der Beitrag aufgrund der Änderung des Integrationsgebietes.



Man erhält durch Anwendung des Gaußschen Satzes als Beitrag aufgrund der Änderung des Integrationsgebietes:

$$\lim_{\Delta t\to o}\frac{1}{\Delta t}\oint_{\partial\Omega_t}\underline{v}\Delta t\mathrm{d}A = \int_{\Omega_t}\mathrm{d}\underline{x}\,\underline{\nabla}\cdot(\underline{v}f) \tag{9.3}$$

Massenbilanz

Als Beispiel soll zunächst die Massenbilanz einer Flüssigkeit behandelt werden. Wir wählen dazu als Geschwindigkeit $\underline{v}$ die Flüssigkeitsgeschwindigkeit und wenden das Transporttheorem auf das Integral über ein beliebiges Gebiet an, das sich bedingt durch die Flüssigkeitsbewegung verformt. Aufgrund dieser Definition kann kein Teilchen, das in diesem Gebiet zu einem Zeitpunkt t_0 enthalten war, dieses verlassen; das Integral über die Dichte muß also zu allen Zeiten die enthaltene Masse ergeben und die Änderung des Integrales verschwinden. Damit erhalten wir

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_t}\rho(\underline{x},t)\mathrm{d}\underline{x} = \frac{\mathrm{d}}{\mathrm{d}t}M(\Omega_t) = 0 = \left\{\frac{\partial}{\partial t}\rho + \underline{\nabla}\cdot(\rho\underline{v})\right\}. \tag{9.4}$$

Diese Beziehung gilt unabhängig vom Integrationsgebiet Ω_t und daher auch direkt für die Integranden,

$$\frac{\partial\rho}{\partial t} + \underline{\nabla}\cdot(\rho\underline{v}) = 0. \tag{9.5}$$

Diese Beziehung ist auch als Kontinuitätsgleichung bekannt. Sie führt mit vertauschten Rollen von ρ und c direkt auf die eingangs erwähnte Diffusionsgleichung, wenn wir anstelle des Massenstromes $\underline{v}\rho$ den Diffusionstrom $\underline{j}_D = -D\underline{\nabla}c$ einsetzen.
Ein wichtiger Spezialfall der obigen Beziehung ergibt sich für eine konstante Massendichte, wie sie für viele Flüssigkeiten und Gase bei nur langsam veränderlichen

Bewegungszuständen angenommen werden kann. Für solche inkompressiblen Stoffe gilt dann,

$$\frac{\partial \rho}{\partial t} + \rho \underline{\nabla} \cdot \underline{v} = \rho \underline{\nabla} \cdot \underline{v} = 0, \tag{9.6}$$

oder noch einfacher,

$$\underline{\nabla} \cdot \underline{v} = 0. \tag{9.7}$$

Impulsbilanz

Der Impuls aller im Gebiet Ω_t enhaltenen Teilchen ergibt sich aus dem Integral über das orts- und zeitabhängige Produkt aus Dichte und Geschwindigkeit,

$$\underline{p}(t) = \int_{\Omega_t} \rho(\underline{x},t)\underline{v}(\underline{x},t)\,\mathrm{d}\underline{x}. \tag{9.8}$$

Das 2. Newtonsche Axiom $\underline{F} = \dot{\underline{p}}$ sagt uns, daß sich der Impuls aufgrund der Kräfte ändert, die an dem Gebiet angreifen. Diese können wir in 2 Anteile aufspalten, nämlich einmal in Volumenkräfte (V) wie die Gravitation, oder Trägheitskräfte, die an jedem Punkt wirksam werden können und in kurzreichweitige Kontakt- oder Flächenkräfte (A), die über die Oberfläche der Elemente wirksam werden können,

$$\frac{\mathrm{d}}{\mathrm{d}t}\underline{p}(t) = \sum \text{Kräfte} = \underbrace{\sum_{\Omega} \underline{F}_V}_{\text{(V)}} + \underbrace{\sum_{\partial\Omega} \underline{F}_A}_{\text{(A)}}. \tag{9.9}$$

Die Summe der Volumenkräfte ergibt sich als einfaches Integral über das Volumen des Gebietes, die Oberflächenkräfte müssen durch ein Oberflächenintegral erhalten werden über infinitesimale vektorielle Kraftelemente $\mathrm{d}\underline{f}$,

$$\text{(V)} = \int_{\Omega_t} \mathrm{d}\underline{x}\, \underline{f}_V, \tag{9.10}$$

$$\text{(A)} = \int_{\partial\Omega_t} \mathrm{d}\underline{f}_A. \tag{9.11}$$

Wenn wir $\partial\Omega$ in sehr kleine vektorielle Oberflächenelemente $\mathrm{d}\underline{A}$ aufteilen, so können wir zwischen $d\underline{f}$ und $d\underline{A}$ einen homogenen, linearen Zusammenhang der Form $d\underline{f} = d\underline{A} \cdot \underline{\underline{\sigma}}$ ansetzen.
Die Komponenten der Größe $\underline{\underline{\sigma}}$ kann man sich als eine Matrix vorstellen, die auf die Komponenten des Vektors des Flächenelementes mit Hilfe der Matrixmultiplikation angewandt wird. Im Spezialfall von reinen Druckkräften ist diese Größe $\underline{\underline{\sigma}}$ diagonal, d.h. daß für ihre (Matrix)Komponenten $\sigma_{ij} = -p\delta_{ij}$ gilt. Das negative Vorzeichen kommt daher, daß die diesem Druck entsprechende Kraft $\mathrm{d}\underline{f}$ dann positiv gezählt wird, wenn sie in das *Innere* des Gebietes gerichtet ist, während die Flächennormale immer

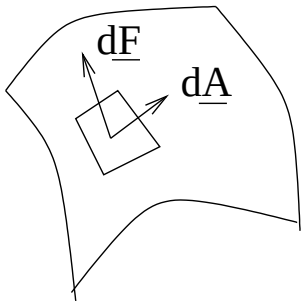


Abbildung 9.1: Der Spannungstensor vermittelt den allgemeinst möglichen linearen Zusamenhang zwischen den vektoriellen Kräften $d\underline{f}$ und den Flächenelementen $d\underline{A}$ der Integrationsoberfläche

nach *außen* zeigt. Diese Proportionalität von Druckkraft und Fläche ist die bereits aus der Schulphysik bekannte.
Der *Spannungstensor* $\underline{\underline{\sigma}}$ muß bestimmte Bedingungen erfüllen, damit sein Produkt mit $d\underline{A}$ die Eigenschaften einer physikalischen Kraft hat. Ohne weitere Herleitung geben wir von diesen die folgenden hier an:

1. Um die Transformationseigenschaften einer Kraft bei Wechsel des Koordinatensystems zu erfüllen, muß $\underline{\underline{\sigma}}$ ein *Tensor* sein, d.h. die Komponenten transformieren sich wie die z.B. die Komponenten des Trägheits*tensors*

2. Aus der Forderung, daß Drehmomente endlich bleiben sollen, folgt bei Abwesenheit von “Volumendrehmomenten” die Symmetrie des Spannungstensors (siehe Landau–Lifschitz, Fluidmechanik, [2])

$$\sigma_{ij} = \sigma_{ji}$$

3. Für Flüssigkeiten kann man den Spannungstensor ausdrücken als eine Funktion des lokalen Druckes und der lokalen Geschwindigkeiten bzw. deren Ableitungen.

   (a) Die Bewegung idealer Flüssigkeiten oder Gase wird nur durch Trägheitskräfte und konservative Druckkräfte kontrolliert:

   $$\sigma_{ij} = -p\delta_{ij}.$$

   (b) Will man Reibung in Flüssigkeiten berücksichtigen, so kann diese nur von Geschwindigkeitsgradienten innerhalb der Flüssigkeit abhängig sein, aber nicht von der absoluten Geschwindgkeit selbst, denn diese läßt sich ja durch eine geeignete Galilei-Transformation immer lokal eliminieren, während Reibungskräfte koordinatensystemunabhängig sein müssen. Weiterhin benötigen wir einen symmetrischen Ausdruck in den Geschwindigkeitskomponenten und den Koordinaten nach denen die Ableitungen gebildet werden,

um die Symmetrieeigenschaften von $\underline{\underline{\sigma}}$ sicherstellen zu können. Für inkompressible, viskose Flüssigkeiten (oder Gase) gilt

$$\sigma_{ij} = -p\delta_{ij} + \eta\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right).$$

Wir betrachten jetzt nochmals das Oberflächenintegral (9.11). Durch Anwendung des Gaußschen Satzes läßt es sich auf ein Volumenintegral über die Divergenz der Spannungen

$$\int_{\partial\Omega_t} \mathrm{d}\underline{f} = \int_{\partial\Omega_t} \mathrm{d}\underline{A}\cdot\underline{\underline{\sigma}}\mathrm{d}\underline{A} = \int_{\Omega_t} \mathrm{d}\underline{x}\,\underline{\nabla}\cdot\underline{\underline{\sigma}} \tag{9.12}$$

zurückführen. Die linken Seite von Gl. (9.9) wollen wir durch (9.8) ersetzen und dann komponentenweise das Reynoldsche Transporttheorem anwenden. Hierzu überlegen wir uns zunächst die Divergenz eines Vektor-Skalar-Produktes

$$\begin{aligned}\underline{\nabla}\cdot(s\underline{v}) &= (\underline{\nabla}s)\cdot\underline{r} + s\cdot(\underline{\nabla}\cdot\underline{v}) \\ &= (\underline{v}\cdot\underline{\nabla})\,s + s\,(\underline{\nabla}\cdot\underline{v}).\end{aligned} \tag{9.13}$$

Damit erhalten wir für die zeitliche änderung des Impulses im mitbewegten Gebiet Ω_t

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}x}\underline{\rho} &= \frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega \mathrm{d}\underline{x}\rho\,(\underline{x},t)\,\underline{v}\,(\underline{x},t) \\ &= \int_{\Omega_t} \mathrm{d}\underline{x}\frac{\partial}{\partial t}\left(\rho\,(\underline{x},t)\,\underline{v}\,(\underline{x},t)\right) + \int_\Omega \mathrm{d}\underline{x}\begin{pmatrix}\underline{\nabla}\cdot(v_x\,\rho\,(\underline{x},t)\,\underline{v}\,(\underline{x},t)) \\ \underline{\nabla}\cdot(v_y\,\rho\,(\underline{x},t)\,\underline{v}\,(\underline{x},t)) \\ \underline{\nabla}\cdot(v_z\,\rho\,(\underline{x},t)\,\underline{v}\,(\underline{x},t))\end{pmatrix} \qquad (9.14) \\ &= \int_{\Omega_t} \mathrm{d}\underline{x}\frac{\partial}{\partial t}\left(\rho\,(\underline{x},t)\,\underline{v}\,(\underline{x},t)\right) \\ &\quad + \int_\Omega \mathrm{d}\underline{x}\,(\underline{v}\cdot\underline{\nabla})\left(\rho\,(\underline{x},t)\,\underline{v}\,(\underline{x},t)\right) + \rho\,(\underline{x},t)\,\underline{v}\,(\underline{x},t)\,(\underline{\nabla}\cdot\underline{v}) \qquad (9.15)\end{aligned}$$

Jetzt sammeln wir alle Ausdrücke in (9.9) und nutzen wieder aus, daß Ω_t beliebig ist, wir also die Beziehung für die Integrale durch eine solche für die Integranden ersetzen können

$$\frac{\partial}{\partial t}(\rho\underline{v}) + \rho\underline{v}\,(\nabla\cdot\underline{v}) + (\underline{v}\cdot\underline{\nabla})\,(\rho\underline{v}) = \underline{f} + \underline{\nabla}\cdot\underline{\underline{\sigma}} \tag{9.16}$$

Die Inkompressibilität in der Form von Gl. 9.7, d.h. $\underline{\nabla}\cdot\underline{v} = 0$ führt uns zu den noch dimensionsabhängigen *Navier-Stokes-Gleichungen*

$$\begin{aligned}\rho\frac{\partial}{\partial t}\underline{v} + \rho\,(\underline{v}\cdot\underline{\nabla})\,\underline{v} &= \underline{\nabla}\cdot\underline{\underline{\sigma}} + \underline{f} \\ &= -\nabla p + \eta\nabla^2\underline{v} + f\end{aligned}$$

Wir haben bisher gesehen, daß wir mit Hilfe des Transporttheorems im wesendlichen Differentialgleichungen für Erhaltungsgrößen herleiten konnten, wie die Kontinuitätsgleichung für die Erhaltung der Masse und die Navier-Stokes-Gleichungen als Bilanzgleichungen für den Impuls.
Nicht betrachtet haben wir die Gleichungen, die uns die Energieerhaltung liefern würde. Hier ginge die Produktion von Wärme durch die Dissipation in dem Fluid ein, die Wärmeleitung, sowie gegebenenfalls die Randbedingungen an diese Gleichung. Wenn die Wärmeausdehungskoeffizienten klein, bzw. nur langsame Strömungen und keine äußeren Wärmequellen vorhanden sind, dann kann man häufig die Wärmeeffekte vernachlässigen. Deren Berücksichtigung führt sonst auf komplexere Gleichungen, in denen die Wärmeausdehnung berücksichtigt werden muß. Wir wollen im folgenden solche Effekte vernachlässigen.
Die Form der Gleichung (9.17) läßt sich noch etwas vereinfachen. Dazu dividieren wir durch ρ, und verwenden das Symbol p jetzt für p/ρ, sowie $\underline{f}$ für die dichtebezogene Kraftdichte $\underline{f}_V/\rho$. Weiterhin führen wir ν für die kinematische Viskosität ν/ρ ein, also

$$\frac{\partial}{\partial t}\underline{v} + (\underline{v}\cdot\underline{\nabla})\,\underline{v} = -\underline{\nabla}p + \nu\nabla^2\underline{v} + \underline{f} \tag{9.17}$$

Es soll hier auch noch die sehr ähnliche (und in weitergehenden Betrachtungen sehr bequeme) dimensionslose Form der Gleichungen angegeben werden. Hierzu bezieht man die Geschwindigkeit und alle Längen auf Referenzwerte U und L und dividiert die Gleichung durch U^2/L, d.h. formal durch eine Beschleunigung. Beachtet man weiterhin die Ersetzungen

$$v = v^* U, \quad \frac{\partial}{\partial t} = \frac{U}{L}\frac{\partial}{\partial t^*}, \quad p^* = \frac{1}{\rho U^2}p, \tag{9.18}$$

wobei p wieder der physikalische Druck aus der Beziehung (9.17) ist, dann erhält man die Gleichung

$$\frac{\partial v^*}{\partial t^*} + \underline{v}^*\cdot\underline{\nabla}^*\underline{v}^* = -\nabla^* p^* - \frac{1}{Re}\underline{\nabla}^{*2}\underline{v}^* + \underline{f}^* \tag{9.19}$$

Hier taucht als einziger physikalischer Parameter des Problems die sogenannte Reynoldszahl $Re = U^2 v/\nu$ auf. Man sieht sofort die Ähnlichkeit zu (9.17), in der die auftretenden Größen noch dimensionsbehaftet sind. Vorteil der dimensionslosen Formulierung ist häufig die minimale Anzahl der physikalischen Parameter des Systems, und weiterhin, daß bei passender Wahl von U und L alle vorkommenden Zahlenwerte in der Größenordnung 1 liegen, was manchmal zur Reduktion numerischer Rundungsfehler beitragen kann.
Im folgenden wollen wir uns auf die noch dimensionsbehaftete Form (9.17). Aufgrund der sogenannten konvektiven Nichtlinearität, dem Term $\underline{v}\cdot\underline{\nabla v}$, ist die Navier-Stokes-Gleichung in den allermeisten Fällen nicht analytisch behandelbar und wir sind auf numerische Lösungsmethoden angewiesen, die wir im folgenden behandeln wollen.

Explizite Zeitdiskretisierung

Wir bringen zunächst den konvektiven Term auf die rechte Seite der Gleichung und ersetzen die partielle Zeitableitung durch einen Differenzenquotienten, gebildet mit einem Zeitschritt Δt,

$$\frac{\partial \underline{v}(t)}{\partial t} \approx \frac{\underline{v}^{n+1} - \underline{v}^n}{\Delta t}. \tag{9.20}$$

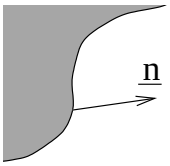
An allen anderen Stellen ersetzen wir die Geschwindigkeit durch ihre alten Werte v^n, im ersten Zeitschritt etwa die bekannten Anfangsgeschwindigkeiten überall im Raum. Eine solche Diskretisierung heißt *explizit*, weil die gesuchte Geschwindigkeit zur neuen Zeit v^{n+1}, die numerische Näherung der exakten Lösung
$v(t_0 + (n+1)\Delta t)$ ist, nur an einer Stelle auftaucht, und die Gleichung daher nach dieser Variablen umgeformt werden kann. Ein Zeitschritt erfordert daher nur sehr geringen Rechenaufwand, nämlich die Auswertung der rechten Seite der diskretisierten Gleichung und dann einen Schritt, um die "alte" zur "neuen" Geschwindigkeit zu machen.

$$\frac{\underline{v}^{n+1} - \underline{v}^n}{\Delta t} = -\underline{\nabla} p^{n+1} + \nu \nabla^2 \underline{v}^n + f^n - \underline{v}^n \cdot \underline{\nabla} \underline{v}^n \tag{9.21}$$

Das obige Schema scheint auf den ersten Blick einleuchtend, hat aber das Problem, daß wir noch nichts über den Druck ausgesagt haben. Dieser muß sich ja irgendwie auf die neue Strömungssituation einstellen. Wir haben aber noch gar keine Gleichung, die uns erlaubte, einen "neuen" Druck zu berechnen. Da wir bisher die Inkompressibilitätsbedingung nicht benutzt haben, liegt es nahe, diese auf Gleichung (9.21) anzuwenden und unter der Annahme $\underline{\nabla} \underline{v}^{n+1} = \underline{\nabla} \underline{v}^n = 0$ eine Bedingung an das "neue" Druckfeld p^{n+1} zu erhalten.

$$0 = -\nabla^2 p^{n+1} + \nu \nabla^2 (\underline{\nabla} \cdot \underline{v}^n) + \underline{\nabla} \cdot \underline{f}^n - \underline{\nabla} (\underline{v}^n \cdot \underline{\nabla}) \underline{v}^n \tag{9.22}$$

Diese Gleichung ist eine Poisson-Gleichung für den neuen Druck p^{n+1}. Wir wissen jetzt, daß wir eine solche Gleichung nur dann eindeutig lösen können, wenn wir Randbedingungen an das Druckfeld angeben, etwa die ersten Ableitungen von p am Rand oder aber direkt die Werte von p. Um solche Bedingungen zu erhalten, projiziert man häufig Gleichung (9.21) auf den Rand des Gebietes. Formal kann man etwa mit der Randnormalen $\underline{n}$ multiplizieren. Der Druckgradient verwandelt sich dann in den Ausdruck $(\underline{n} \cdot \underline{\nabla}) p$, den wir auch als Richtungsableitung des Drucks $\partial p / \partial n$ in Richtung der nach außen zeigenden Flächennormalen schreiben können. Die Auswertung der anderen Terme ergeben dann den Wert dieser Ableitung. Allerdings ist diese Auswertung aufgrund der Ortshängigkeit von $\underline{n}$ der im allgemeinen am Rand nicht verschwindenden zweiten Ableitungen der Normalkomponente der Flüssigkeitsgeschwindigkeit nur wieder numerisch möglich.
Wir wollen im folgenden ein Schema angeben, das die Berechnung dieser Ableitungen nicht erfordert. Es soll jedoch an dieser Stelle darauf hingewiesen werden, daß die Auswertung im allgemeinen Fall erforderlich ist.



$$\frac{\partial p}{\partial n} = (\underline{n} \cdot \underline{\nabla})\, p$$

Operator-Aufspaltungs-Technik

Wir beginnen mit einer Aufspaltung der Zeitableitung in zwei Schritte, indem wir eine Zwischenvariable $\underline{v}^*$ einführen, die keine physikalische Bedeutung hat, sondern nur rein numerisch relevant ist,

$$\frac{v^{n+1} - v^* + v^* - v^n}{\Delta t} = -\nabla p^{n+1} + \nu \nabla^2 \underline{v}^n + f^n - \underline{v}^n \cdot \underline{\nabla v}^n. \tag{9.23}$$

Diese Gleichung spalten wir jetzt in zwei auf,

$$\frac{v^* - v^n}{\Delta t} = -\nu \nabla^2 \underline{v}^n + f^n - (\underline{v}^n \cdot \underline{\nabla})\, \underline{v}^n \tag{9.24}$$

$$\frac{v^{n+1} - v^*}{\Delta t} = -\nabla p^{n+1} \tag{9.25}$$

Die Druckgleichung erhalten wir wieder durch die Anwendung des Divergenzoperators, jetzt auf (9.25),

$$\nabla^2 p^{n+1} = \frac{\nabla \cdot v^*}{\Delta t}. \tag{9.26}$$

Aus (9.25) erhalten wir auch die Projektion auf die Randnormale, die uns die Druckrandbedingungen liefern muß,

$$\frac{\partial p}{\partial n} = \frac{1}{\Delta t} \underline{n} \cdot (\underline{v}^{n+1} - \underline{v}^*). \tag{9.27}$$

Räumliche Diskretisierung

Wir müssen uns jetzt, um diese Gleichungen weiter auswerten zu können, der räumlichen Diskretisierung des Problems widmen. Dazu führen wir ein versetztes Gitter (*staggered grid* oder auch Marker-and-Cell-Gitter) ein. Vektorielle Variablen wie die Kräfte und Geschwindgkeiten sitzen in diesem Gitter auf den Mitten der Zellwände, während die skalaren Variablen in Zentrum jeder Zelle sitzen.
Die einfachste Diskretierung ist die der skalaren Poissongleichung (9.26), bei der die bekannte Sternform des Laplace-Operators auftritt und die Divergenz der vektoriellen Hilfsvariable $\underline{v}^*$ approximiert werden muß. Für den auch in (9.25) auftauchenden

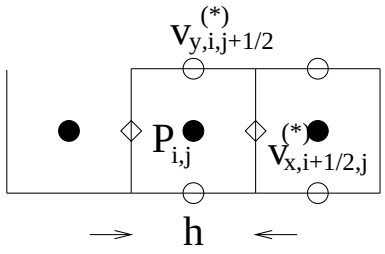


Abbildung 9.2: versetztes Gitter *(staggered grid):* Die Komponenten der Geschwindigkeit sind um eine halbe Gitterkonstante in die jeweilige Raumrichtung verschoben und sitzen auf den Zellwänden, zu denen sie normal sind. Skalare Variablen sind mit der Zellmitte assoziiert. Die Verschiebung deuten wir durch halbzahlige Indizes an.

Druckgradienten und den auf den Druck angewandten Laplace-Operator finden wir z.B.:

$$(\nabla p)_j = \left(p_{i+1,j} - p_{i,j}\right)/h, \tag{9.28}$$

$$\underline{\nabla}^2 p = \frac{1}{h^2}\left(p_{i+1,j} + p_{i-1,j} + p_{i,j+1} - p_{i,j-1} - 4p_{i,j}\right). \tag{9.29}$$

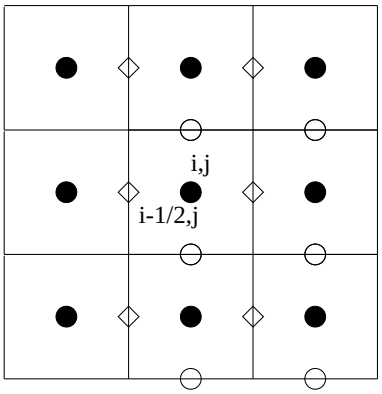
Hier haben wir h als Gitterkonstante der Diskretisierung verwendet. Für den von $\underline{v}^*$ abhängigen Quellterm erhalten wir

$$\underline{\nabla}\cdot\underline{v}^* = \frac{1}{h}\left(v^*_{x,i+\frac{1}{2},j} - v^*_{x,i-\frac{1}{2},j} + v^*_{y,i,j+\frac{1}{2}} - v^*_{y,i,j-\frac{1}{2}}\right) \tag{9.30}$$

Man sieht an diesem Beipiel, wie schön einerseits eine zentrierte Ableitung der Geschwindigkeiten mit dem skalaren Druck “räumlich” zusammenfällt und andererseits wiederum eine räumliche Ableitung des Druckes gerade mit den vektoriellen Geschwindigkeitsvariablen. Da wir bereits wissen, daß zentrierte Ableitungen von zweiter Ordnung genau sind, ist vielleicht klar, wie hier diese geschickte Anordnung der Variablen in natürlicher Weise zu einer räumlichen Diskretisierung von zweiter Ordnung führt.
Die Diskretisierung des konvektiven Terms gestaltet sich etwas aufwendiger, denn nicht alle Variablen befinden sich bereits an passenden Orten. Vielmehr muß man sich einige durch Interpolation von Nachbarwerten erst passend erzeugen.
Die Form $(\underline{v}\cdot\underline{\nabla})\,\underline{v}$ bezeichnet man häufig als die nichtkonservative Form des konvektiven Terms. Es gibt auch demzufolge noch eine konservative Form, die wir weiter unten kennenlernern werden. Wir müssen eine Form der Diskretisierung finden, die die konvektive Ableitung z.B. der x-Komponente der Geschwindigkeit

$$(\underline{v}\cdot\underline{\nabla})\,v_x = v_x(\partial/\partial x)v_x + v_y(\partial/\partial y)v_x$$

auch wieder am Ort der x-Komponente der Geschwindigkeit ergibt, denn dort muß die Zeitableitung eben dieser Komponente gebildet werden. Diese Forderung führt z.B.



auf die Form

$$v_{x,i+1/2,j} \cdot \frac{1}{2h}\left(v_{x,i+3/2,j} - v_{x,i-1/2,j}\right) \quad + \frac{1}{4}\left(v_{y,i,j+1/2} + v_{y,i,j-1/2} + v_{y,i+1,j+1/2} + v_{y,i+1,j-1/2}\right) \times \frac{1}{2h}\left(v_{x,i+1/2,j+1} + v_{x,i+1/2,j-1}\right)$$

Ihr Nachteil ist, daß die neu zu berechnende Geschwindigkeitskomponente $v^x_{i+1/2,j}$ nur an einer Stelle in die Diskretisierung eingeht und beispielsweise nicht in die Berechnung der räumlichen Ableitungen. Dies wird bei der konservativen Form

$$(\underline{\nabla} \cdot \underline{\nabla})\,\underline{v} = \underline{\nabla}\,(\underline{v}\,\underline{v}) \tag{9.31}$$

der konvektiven Ableitung vermieden, die allerdings die Inkompressibilität der Strömung voraussetzt,

$$\sum_i \frac{\partial}{\partial x_i} v_i v_j = \sum_i \left(\frac{\partial}{\partial x_i} v_i\right) v_j + \sum_i \left(v_i \partial_i v_j\right) \tag{9.32}$$

$$= (\underline{v} \cdot \underline{\nabla})\, v_j. \tag{9.33}$$

$$\tag{9.34}$$

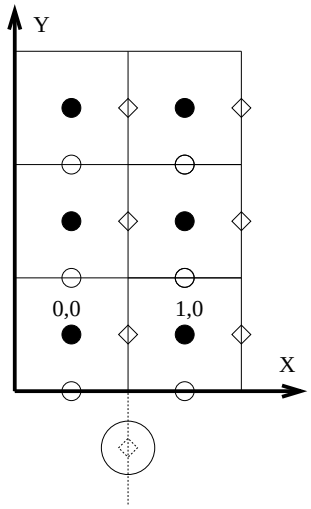
Der Ausdruck $(\underline{v}\,\underline{v})$ ist dabei ein Tensor, dessen kartesische Komponenten die Produkte der Geschwindigkeitskomponenten sind, was wir hier etwas lax unter Benutzung des Gleichheitszeichens schreiben wollen:

$$(\underline{v}\,\underline{v}) = \begin{pmatrix} v_x v_x & v_x v_y \\ v_y v_x & \ddots \end{pmatrix}. \tag{9.35}$$

Die Diskretisierung dieses Ausdruck überlassen wir dem interessierten Leser als Verständnisübung. Sie findet sich in Ref. [1].
Die Diskretisierung des sogenannten viskosen Terms $\nabla^2 \underline{v}$ sollte keine Schwierigkeiten bereiten, da sie bereits für die Komponenten des Druckes oben durchgeführt wurde.

Diskretisierung der Randbedingungen

Nachdem wir die in der Differentialgleichung (9.24) auftretenden Terme diskretisiert haben, müssen wir uns noch Gedanken um die Randbedingungen machen. Für die



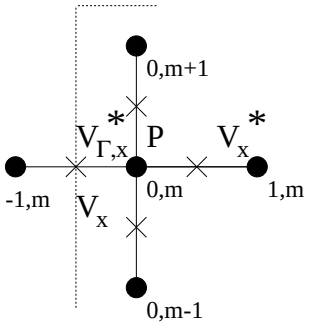
Zu einer vereinfachten Behandlung der Randbedingungen werden noch Komponenten jenseits des eigentlichen Randes eingeführt, die so bestimmt werden, daß für die dort auszuwertenden Differentialoperatoren die Randbedingungen erfüllt sind.

Navier-Stokes-Gleichung mit festen Rändern verwendet man üblicherweise Haftrandbedingungen (*no-slip Bedingung*), bei denen angenommen wird, daß am Rand die Geschwindigkeit der Flüssigkeit und die des Randes übereinstimmen. Aufgrund der Lage der Geschwindigkeitskomponenten auf dem Gitter ist klar, daß nur eine Komponente direkt auf dem Rand definiert ist, während für die (beiden in 3D) andere(n) eine Interpolationsprozedur gefunden werden muß. Hier findet als einfachste Möglichkeit ein Spiegelungsprinzip Anwendung, bei dem jenseits des Randes (negativer Index) noch ein Hilfswert definiert und durch die Bedingung

$$\frac{1}{2}\left(v_{x,i,-1/2} + v_{x,i,1/2}\right) = v_{x,\partial\Omega} \tag{9.36}$$

festgelegt wird. Obwohl diese Bedingung am Rand nicht von der gleichen Genauigkeit wie die sonstige Diskretisierung ist, führt sie in der Praxis doch zu guten Resultaten. Näherungen höherer Ordnung findet man in [1].

Die Diskretisierung auf einem so versetzten Gitter hat auch den Vorteil, daß sich die Druckrandbedingungen sehr einfach formulieren lassen.



Poisson-Gleichung

$$\Delta p^{n+1} = \frac{\nabla \underline{v}^*}{\Delta t} \quad \text{(I)}$$

Neumann-Randbedingungen

$$\left(\frac{\partial p}{\partial \underline{n}}\right)^{n+1}_{\partial\Omega} = \frac{1}{\Delta t}\left(\underline{v}^{n+1}_{\partial\Omega} - \underline{v}^*_{\partial\Omega}\right)\cdot \underline{n} \quad \text{(II)}$$

Schauen wir uns etwa die Gleichungen an, die durch Diskretisierung der Poisson-Gleichung für den Druck und die Neumann-Randbedingungen entstehen:

$$\begin{aligned}\text{(I)} \qquad & \frac{1}{h}\left(\frac{p^{n+1}_{1,m} - p^{n+1}_{0,m}}{h} - \frac{p^{n+1}_{0,m} - p^{n+1}_{-1,m}}{h}\right) \\ & + \frac{1}{h}\left(\frac{p^{n+1}_{0,m+1} - p^{n+1}_{0,m}}{h} - \frac{p^{n+1}_{0,m} - p^{n+1}_{0,m-1}}{h}\right) \\ & = \frac{1}{\Delta t}\left(\frac{v^*_{x,1/2,m} - v^*_{x,\Gamma}}{h} + \frac{v^*_{y,0,m+1/2} - v^*_{y,0,m-1/2}}{h}\right) \qquad (9.37)\end{aligned}$$

$$\text{(II)} \qquad \frac{1}{h}\left(p^{n+1}_{0,m} - p^{n+1}_{-1,m}\right) = \frac{1}{\Delta t}\left(v^{n+1}_{x,\Gamma} - v^*_{x,\Gamma}\right) \qquad (9.38)$$

Durch Einsetzen der Randbedingung in die diskretisierte Laplace-Gleichung am Rand sieht man, daß deren Lösung gar nicht vom Wert der Variablen $\underline{v}^*$ am Rand abhängig sein kann, denn dieser hebt sich gerade auf beiden Seiten der Gleichung heraus. Damit können wir über deren Wert frei verfügen und insbesondere $v^*_{x,\partial\Omega} = v^{n+1}_{x,\partial\Omega}$ setzen. Damit wird dann die Randbedingung an den Druck sehr einfach und besagt, daß die Normalenableitung über den Rand verschwindet,

$$p^{n+1}_{0,m} - p^{n+1}_{-1,m} = 0. \qquad (9.39)$$

Wir haben jetzt alle Elemente zusammen, um den Lösungsvorgang für einen Zeitschritt der Diskretisierung der Navier-Stokes-Gleichung zu formulieren. Wir bestimmen zunächst unter Berücksichtigung der Geschwindigkeitsrandbedingungen den Wert von v^* aus Gleichung (9.24). Danach lösen wir die Poisson-Gleichung des Druck-Problems mit verschwindenden Ableitungen am Rand. Als letztes verbleibt, mit dem so gefundenen neuen Druck in die Gleichung (9.25) einzugehen und dort den neuen Wert der Geschwindigkeit zu berechnen.

Literatur

[1] Peyret Taylor, *Computational Methods for Fluid Flow*, (Springer 1983).
[2] Landau, Lifshitz, Lehrbuch der Theoretischen Physik, Band 6, *Fluidmechanik*, Akademie Verlag, Berlin.

9.9 Finite Elemente

Die Methode der finiten Elemente wurde von Ingenieuren erfunden und entwickelt. Der Name stammt von Clough (1960). Gegeben sei ein Gebiet auf dem die Werte einer Funktion $\Phi(\vec{r})$ so bestimmt werden sollen, daß sie vorgegebene Randbedingungen erfüllen. Dazu trianguliert man das Gebiet zunächst, d.h. man zerlegt es in beliebige, nicht notwendig gleich große Dreiecke. Ein einzelnes Dreieck nennt man ein Element. Im Unterschied zu den bisher besprochenen Methoden betrachtet man die Werte von Φ nicht auf den Gitterpunkten, sondern auf einer über dem Element aufgespannten Fläche, z.B. einer Ebene

$$\Phi(\vec{r}) = c_1 + c_2 x + c_3 y.$$

Die Koeffizienten c_i sind Funktionen der Werte von Φ auf den Eckpunkten des Elements. Es ist leicht einsichtig, daß die so aus Dreiecken aufgebaute Fläche stetig ist.
Ein allgemeineres Verfahren nähert Φ durch Paraboloide

$$\Phi(\vec{r}) = c_1 + c_2 x + c_3 y + c_4 x^2 + c_5 xy + c_6 y^2$$

an. Die Koeffizienten c_i ergeben sich hier aus den Werten von Φ auf den Eckpunkten der dreieckigen Elemente sowie den drei Mittelpunkten der Seiten. Der Schnitt eines Paraboloids mit dem Rand des Elemente ergibt eine Parabel. Der Verlauf der Parabel ist durch die drei Punkte auf jeder Seite des Elements jedoch eindeutig festgelegt, so daß auch hier die Stetigkeit der Fläche gewährleistet ist.
Man definiert nun ein Funktional

$$E = \iint_G \left(\frac{1}{2} (\nabla\Phi)^2 + \frac{1}{2} a\Phi^2 + b\Phi \right) \mathrm{d}x\, \mathrm{d}y + \int_\Gamma \left(\frac{\alpha}{2} \Phi^2 + \beta\Phi \right) \mathrm{d}s,$$

welches einer Energie entspricht. $\iint_G$ integriert über die Fläche, $\int_\Gamma$ über den Rand eines Elements. Mit einem Variationsprinzip minimiert man dieses Funktional (Argyris 1954).

$$\delta E = \iint_G (\nabla\Phi\delta\nabla\Phi + a\Phi\delta\Phi + b\delta\Phi)\, \mathrm{d}x\, \mathrm{d}y + \int_\Gamma (\alpha\Phi\delta\Phi + \beta\delta\Phi)\, \mathrm{d}s = 0$$

Unter Verwendung des 1.Satzes von Green

$$\iint_G \nabla\Phi\nabla\Psi \mathrm{d}x\, \mathrm{d}y = -\iint_G \Psi\Delta\Phi \mathrm{d}x\, \mathrm{d}y + \int_\Gamma \frac{\partial\Phi}{\partial n}\Psi \mathrm{d}s$$

kann man die letzte Formel auch schreiben als

$$\delta E = \iint_G (-\Delta\Phi + a\Phi + b)\, \delta\Phi \mathrm{d}x\, \mathrm{d}y + \int_\Gamma \left(\alpha\Phi + \beta + \frac{\partial\Phi}{\partial n} \right) \delta\Phi \mathrm{d}s = 0.$$

Damit diese Gleichung erfüllt ist, muß jedes Integral einzeln gleich Null sein. Das erste Integral führt dann auf die Gleichung

$$\Delta\Phi = a\Phi + b.$$

Für $a = 0$ ist das aber genau die Poissongleichung. Für $b = 0$ erhält man die Helmholtzgleichung.
Am Beispiel des ersten Terms der Gesamtenergie

$$E = \sum_{\text{Elemente}} \left(\iint_{G_i} (\nabla\Phi)^2 + a \iint_{G_i} \Phi^2 + b \iint_{G_i} \Phi \right)$$

soll nun gezeigt werden, wie man dieses Integral auf die Form

$$E = \vec{\Phi}\mathbf{A}\vec{\Phi} + \vec{b}\Phi$$

bringen kann. Die Minimalisierung von E liefert dann nämlich

$$\frac{\partial E}{\partial \Phi} = 0 \qquad \Rightarrow \qquad \mathbf{A}\vec{\Phi} + \vec{b} = 0,$$

und somit die bekannte Gleichung, zu deren Lösung auf die bereits vorgestellten Verfahren zurückgegriffen werden kann.

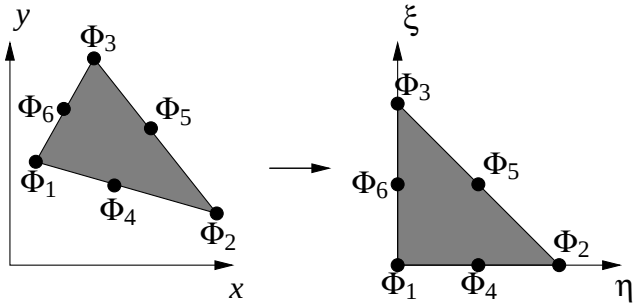


Abb. 9.4 *Transformation eines beliebigen Elements auf das Standardelement*

Mit Hilfe der Transformation

$$\begin{aligned} x &= x_1 + (x_2 - x_1)\xi + (x_3 - x_1)\eta \\ y &= y_1 + (y_2 - y_1)\xi + (y_3 - y_1)\eta, \end{aligned}$$

also

$$\begin{aligned} \eta &= \left((y - y_1)(x_2 - x_1) - (x - x_1)(y_2 - y_1) \right) / D \\ \xi &= \left((x - x_1)(y_3 - y_1) - (y - y_1)(x_3 - x_1) \right) / D \end{aligned}$$

mit $D = (y_3 - y_1)(x_2 - x_1) - (y_2 - y_1)(x_3 - x_1)$ transformiert man das Element G zunächst auf ein "Standardelement" T. Die Koordinaten des Standardsystems seien ξ und η. In diesem neuen Koordinatensystem erhält $\nabla\Phi = \left(\frac{\partial\Phi}{\partial x}, \frac{\partial\Phi}{\partial y} \right)$ das Aussehen

$$\nabla\Phi = \left(\frac{\partial\Phi}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial\Phi}{\partial\eta}\frac{\partial\eta}{\partial x}, \frac{\partial\Phi}{\partial\xi}\frac{\partial\xi}{\partial y} + \frac{\partial\Phi}{\partial\eta}\frac{\partial\eta}{\partial y} \right).$$

Aus den Transformationsformeln ergibt sich für die Ableitungen

$$\frac{\partial\xi}{\partial x}=\frac{y_3-y_1}{D},\qquad \frac{\partial\xi}{\partial y}=-\frac{x_3-x_1}{D},\qquad \frac{\partial\eta}{\partial x}=-\frac{y_2-y_1}{D},\qquad \frac{\partial\eta}{\partial y}=\frac{x_2-x_1}{D}$$

mit $D=(y_3-y_1)(x_2-x_1)-(y_2-y_1)(x_3-x_1)$. Damit erhält man

$$\begin{aligned}\Phi_x^2 &= \left(\frac{\partial\Phi}{\partial\eta}\frac{\partial\eta}{\partial x}+\frac{\partial\Phi}{\partial\xi}\frac{\partial\xi}{\partial x}\right)^2\\ &= \Phi_\xi^2\frac{(y_3-y_1)^2}{D^2}-2\frac{(y_3-y_1)(y_2-y_1)}{D^2}\Phi_\xi\Phi_\eta+\frac{(y_2-y_1)^2}{D^2}\Phi_\eta^2\\ \Phi_\eta^2 &= \Phi_\xi^2\frac{(x_3-x_1)^2}{D^2}-2\frac{(x_3-x_1)(x_2-x_1)}{D^2}\Phi_\xi\Phi_\eta+\frac{(x_2-x_1)^2}{D^2}\Phi_\eta^2.\end{aligned}$$

Die Indizes x,y,ξ,η bezeichnen hier und im Folgenden die entsprechenden Ableitungen. Der betrachtete Term des obigen Integrals hat im neuen Koordinatensystem damit die Gestalt

$$\iint_G\left(\Phi_x^2+\Phi_y^2\right)\mathrm{d}x\,\mathrm{d}y=\iint_T\left(a_1\Phi_\xi^2+2a_2\Phi_\xi\Phi_\eta+a_3\Phi_\eta^2\right)\mathrm{d}\xi\,\mathrm{d}\eta.$$

Die Koeffizienten a_1,a_2,a_3 ergeben sich aus den beiden Formeln darüber zu

$$\begin{aligned}a_1 &= \frac{(y_3-y_1)^2}{D^2}+\frac{(x_3-x_1)^2}{D^2}\\ a_2 &= 2\frac{(y_3-y_1)(y_2-y_1)}{D^2}+2\frac{(x_3-x_1)(x_2-x_1)}{D^2}\\ a_3 &= \frac{(y_2-y_1)^2}{D^2}+\frac{(x_2-x_1)^2}{D^2}.\end{aligned}$$

Es genügt, diese Koeffizienten einmal für jedes Element zu berechnen und das Ergebnis zu speichern. Alle weiteren Rechnungen können nun auf dem Standardelement durchgeführt werden.
Es soll nun ein Paraboloid so über dem Standardelement aufgespannt werden, daß die gegebenen sechs Werte von Φ auf dem Paraboloid liegen. Dazu entwickelt man Φ gemäß

$$\Phi(\xi,\eta)=\sum_{i=1}^{6}\Phi_iN_i(\xi,\eta)=\vec{\varphi}\vec{N}(\xi,\eta).$$

Φ_i sind die Werte auf den 6 Randpunkten, $(\vec{N})_i=N_i$ sind die Basisfunktionen, die wie folgt gewählt werden:

$$\begin{array}{ll}N_1=(1-\xi-\eta)(1-2\xi-2\eta) & N_4=4\xi(1-\xi-\eta)\\ N_2=\xi(2\xi-1) & N_5=4\xi\eta\\ N_3=\eta(2\eta-1) & N_6=4\eta(1-\xi-\eta)\end{array}$$

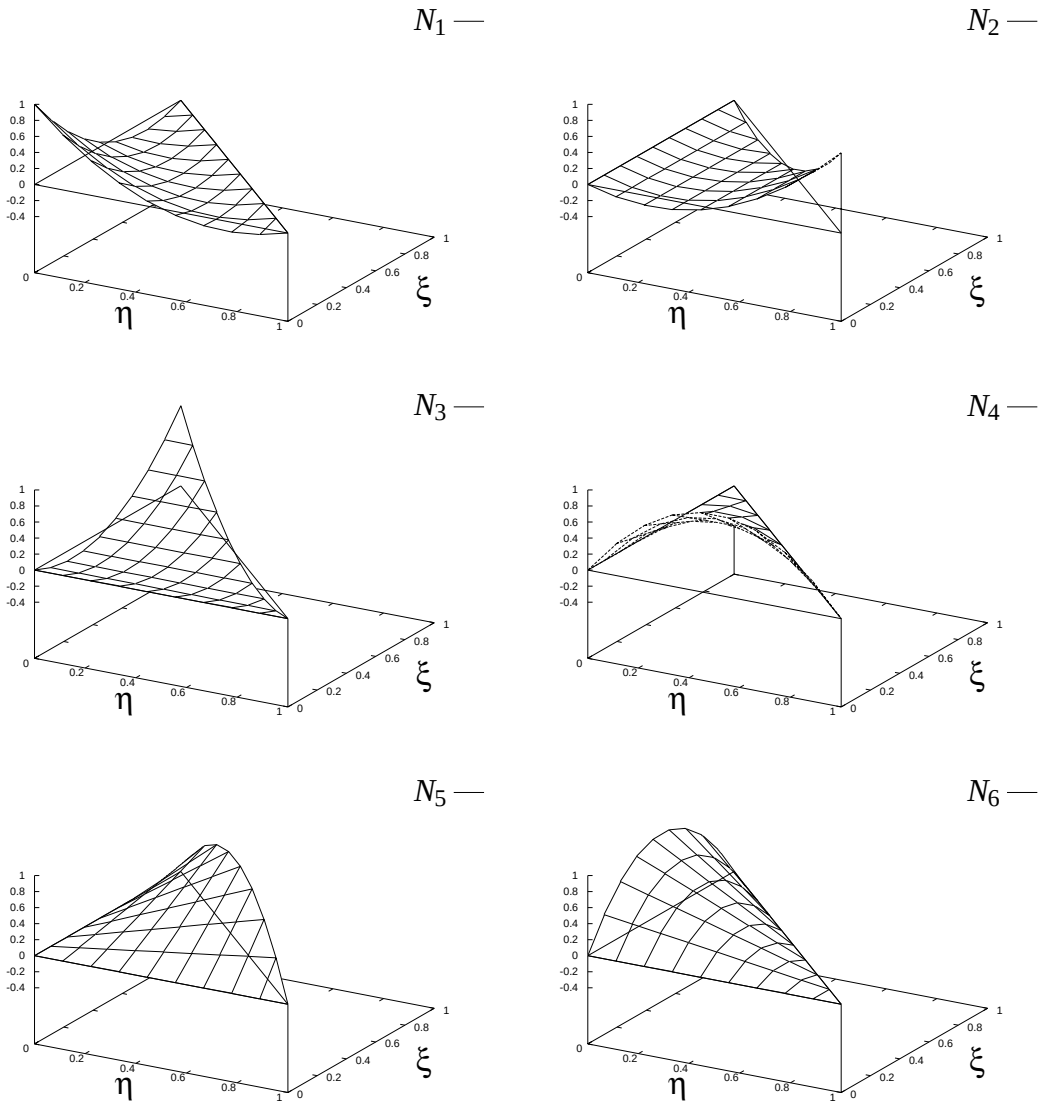


Abb. 9.9.5 *Basisfunktionen zur Entwicklung von* Φ.

Nun ist man soweit, daß man den hier untersuchten Term des Energieintegrals ausrechnen kann.

$$\begin{aligned} I_1 = \iint_T \Phi_\xi^2 \,\mathrm{d}\xi\,\mathrm{d}\eta &= \iint_T \left(\vec{\varphi}\vec{N}_\xi(\xi,\eta)\right)^2 \,\mathrm{d}\xi\,\mathrm{d}\eta \\ &= \iint_T \vec{\varphi}^t \vec{N}_\xi \vec{N}_\xi^t \vec{\varphi} \,\mathrm{d}\xi\,\mathrm{d}\eta \\ &= \vec{\varphi}^t \underbrace{\iint \vec{N}_\xi \vec{N}_\xi^t \,\mathrm{d}\xi\,\mathrm{d}\eta}_{\mathbf{S}_1} \vec{\varphi} \end{aligned}$$

Die Matrix $\mathbf{S}_1$ hat als Elemente nur reine Zahlen, d.h. sie ist unabhängig von ξ und η und muß auch nur einmal ausgerechnet werden. Analog erhält man für die übrigen Terme des Integrals

$$\begin{aligned} I_2 = \iint_T \Phi_\xi \Phi_\eta \, \mathrm{d}\xi \, \mathrm{d}\eta &= \vec{\varphi}^t \mathbf{S}_2 \vec{\varphi} \\ I_3 = \iint_T \Phi_\eta^2 \, \mathrm{d}\xi \, \mathrm{d}\eta &= \vec{\varphi}^t \mathbf{S}_3 \vec{\varphi}. \end{aligned}$$

Für den hier untersuchten ersten Term des Energieintegrals kann man damit abschließend schreiben

$$\iint_G (\nabla \Phi)^2 \, \mathrm{d}x \, \mathrm{d}y = \vec{\varphi}^t \mathbf{S} \vec{\varphi}$$

mit $\mathbf{S} = a_1 \mathbf{S}_1 + 2a_2 \mathbf{S}_2 + a_3 \mathbf{S}_3$. $\mathbf{S}$ bezeichnet man als die *Steifigkeitsmatrix*.
Für die beiden anderen Terme des Energieintegrals muß man die gleichen Umformungen machen. Den zweiten Term kann man dann schreiben als

$$\iint_G a\Phi^2 \, \mathrm{d}x \, \mathrm{d}y = \vec{\varphi}^t \mathbf{M} \vec{\varphi}.$$

$\mathbf{M}$ ist die sogenannte *Massematrix*.

Literaturverzeichnis

[1] Bronstein, Semendjajew, *Taschenbuch der Mathematik*, Verlag Harri Deutsch, Frankfurt/Main.

[2] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry*, North-Holland, Amsterdam (1981).

[3] E. W. Montroll and J. L Lebowitz eds., *Fluctuation Phenomena*, North-Holland, updated paperback edition, Amsterdam (1987).

[4] Daniel Ben Avraham and Shlomo Havlin, *Diffusion in Disordered Media,* Adv. in Physics, **36**, 695–798 (1987).

[5] P. A. P. Moran, *An introduction to probability theory,* Oxford University Press, New York (1984).

[6] F. Reif, *Fundamentals of Statistical and Thermal Physics*, McGraw Hill, Intl. Editions, Singapore, 1985.

[7] A.Bunde, S.Havlin:"Fractals and Disordered Systems", Springer-Verlag, 1991

[8] Malvin H. Kalos and Paula A. Whitlock, *Monte Carlo Methods, Volume 1: Basics*, Wiley, New York (1986).

[9] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth. A.H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).

[10] Kurt Binder and Dieter W. Heermann, *Monte Carlo Simulation in Statistical Physics*, Springer Verlag, Berlin (1992).

[11] I.M.Sobol, *Die Monte-Carlo-Methode*, Verlag Harri Deutsch, Frankfurt (1985)

[12] Kerson Huang, *Statistical Mechanics,* 2nd ed., Wiley (1987).

[13] H. E. Stanley, *Introduction to Phase Transitions and Critical Phenomena,* Oxford University Press (1971).

[14] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, *Numerical Recipes in C*, 2. Auflage, Cambridge University Press, 1992

[15] M.P. Allen, D.J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press, 1991

[16] Friedhelm Kuypers, *Klassische Mechanik*, VCH Verlagsgesellschaft mbH, 1989

[17] J.-P. Hansen, I.R. McDonald, *Theory of simple liquids*, Academic Press, 1990

[18] W.L. Briggs, *A Multigrid Tutorial*, Society for Industrial and Applied Mathematics, 1991