Computational Physics II *

Ulli Wolff
mit
Tomasz Korzec

Institut für Physik der HU
AG Computational Physics

e-mail: uwolff@physik.hu-berlin.de
e-mail: korzec@physik.hu-berlin.de

14. Januar 2014

Zusammenfassung

Die Skripten CP2 (und CP1) aus früheren Semestern finden sich im Web:
`www.physik.hu-berlin.de/com/teachingandseminars/previous_CPI_CPII`

Das zum laufenden Kurs aktualisierte Skriptum wird auf der Webseite zum Kurs fortgeschrieben. Es soll jeweils kurz *vor* der Vorlesung den neuen Stoff enthalten. Bei der Vorbereitung werden dort Korrekturen eingebaut. Auch ändert sich der Stoff bzw. seine Anordnung manchmal. Das aktuelle Skriptum sollte im Zusamenhang mit den Übungen verwendet werden, die alte Ausgabe soll nur einen vorab Eindruck geben.

Danksagung: An der Entstehung dieses Skriptums hat Burkhard Bunk maßgeblichen Anteil. Für die Weiterentwicklung vieler Kapitel danken wir Francesco Knechtli. Material für die Übungen wurde teilweise von Oliver Bär und Tomasz Korzec ausgearbeitet.

*Modul 22.1 Masterstudiengang Physik WS 2013/14

Inhaltsverzeichnis

1 Eigenwerte

Die Notation ist im folgenden die gleiche wie im Kapitel über lineare Gleichungen. Ein Eigenwertproblem für eine quadratische Matrix A besteht aus der Frage nach Vektoren $x \neq 0$ und Zahlen λ, so daß gilt

$$Ax = \lambda x. \tag{1.1}$$

Im allgemeinen Fall sind alle Größen hier komplex. Selbst reelle A können auf komplexe λ führen (paarweise zu einander konjugiert).

1.1 Tatsachen aus der linearen Algebra

Wir wollen einige Resultate referieren, die z. B. in [1] zusammengefaßt sind und in [2] ausführlich diskutiert werden. Aus (1.1) folgt, daß λ genau dann ein Eigenwert ist, wenn gilt

$$P(\lambda) = \det(A - \lambda 1) = 0, \tag{1.2}$$

denn dann hat das lineare System $(A - \lambda 1)x = 0$ nicht verschwindende Lösungen. Bedenkt man die Definition von det, so ist klar, daß P ein Polynom vom Grad n in λ ist für eine $n \times n$–Matrix A, das charakteristische Polynom von A. Dieses hat im Komplexen n Nullstellen, die auch für spezielle Wahlen von a_{ij}, also im nicht generischen, aber doch häufigen Fall zusammenfallen können. Das Auffinden dieser Werte ist i. a. ein transzendentes Problem, zu dem es keine Algorithmen mit fester Zahl von Operationen, wie z.B. bei der Lösung linearer Systeme, geben wird. Stattdessen wird man bestenfalls iterative Methoden wie bei der allgemeinen Nullstellensuche haben, die dann irgendwie konvergieren.

Im generischen Fall gibt es n verschiedene linear unabhängige Eigenvektoren, die also den ganzen Raum aufspannen. Für lauter verschiedene Werte λ sind die Eigenvektoren sogar eindeutig bis auf einen Faktor, der in (1.1) irrelevant ist und somit durch eine Konvention fixiert werden kann. Für beliebige Matrizen und entartete, d.h. zusammenfallende Eigenwerte *kann* es "zu wenige" Eigenvektoren geben. Der Fall ist nicht so exotisch, wie das folgende 2×2 Beispiel zeigt:

$$A = \begin{pmatrix} a & 1 \\ 0 & b \end{pmatrix}, \tag{1.3}$$

$$P(\lambda) = (a - \lambda)(b - \lambda) \ \Rightarrow \lambda_1 = a, \lambda_2 = b. \tag{1.4}$$

Die zugehörigen Eigenvektoren sind trivial zu bestimmen,

$$x^{(1)} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, x^{(2)} = \begin{pmatrix} 1 \\ b-a \end{pmatrix}, \tag{1.5}$$

wo hier die Konvention $x_1 = 1$ verwendet ist. Im Spezialfall $a = b$ gibt es offenbar nur einen Eigenvektor zu einem doppelten Eigenwert.

Wir setzen nun n Eigenvektoren zu $\lambda_1 \dots \lambda_n$ voraus und schreiben diese als Spalten in die Matrix X_R der Rechtseigenvektoren. Dann gilt die Matrixgleichung

$$AX_R = X_R \operatorname{diag}(\lambda_1 \dots \lambda_n), \tag{1.6}$$

wobei $\operatorname{diag}(\lambda_1 \dots \lambda_n)$ die diagonale Matrix aus den Eigenwerten ist. Durch Betrachten des zu (1.1) transponierten Problems

$$yA = \lambda y \tag{1.7}$$

für Zeilen y (gleiche λ !) erhält man auch

$$X_L A = \operatorname{diag}(\lambda_1 \dots \lambda_n) X_L \tag{1.8}$$

Aus beiden Gleichungen zusammen ergibt sich

$$X_L X_R \operatorname{diag}(\lambda_1 \dots \lambda_n) = \operatorname{diag}(\lambda_1 \dots \lambda_n) X_L X_R. \tag{1.9}$$

Die Matrix $X_L X_R$ besteht aus den Skalarprodukten der Links- und Rechtseigenvektoren. Wenn $T = X_L X_R$ mit der Diagonalmatrix der Eigenwerte kommutiert, so heißt das in Komponenten $T_{ij}(\lambda_i - \lambda_j) = 0$. Daraus folgt für den nichtentarteten Fall $\lambda_i \neq \lambda_j$, daß $X_L X_R$ selbst diagonal ist. Die Links- und Rechtseigenvektoren sind orthogonal und können so normiert werden, daß gilt

$$y^{(i)} x^{(j)} = \delta_{ij}. \tag{1.10}$$

Bei entarteten Eigenwerten *kann* der Fall der unvollständigen Eigenvektoren auftreten. Ist dies nicht der Fall, so kann man (1.10) immer noch erreichen. Damit sind dann X_L und X_R invers zueinander, und es gilt

$$\operatorname{diag}(\lambda_1 \dots \lambda_n) = X_R^{-1} A X_R = X_L A X_R \tag{1.11}$$

und A ist ähnlich zur Diagonalmatrix seiner Eigenwerte.

Bisher wurden allgemeine komplexe Matrizen betrachtet. Wir wollen nun zunehmend spezialisieren. Wenn gilt

$$A^\dagger A = AA^\dagger, \tag{1.12}$$

so heißt eine Matrix normal, wobei die adjungierte Matrix $A^\dagger$ tansponiert und komplex konjugiert ist. Solche Matrizen haben immer ein vollständiges System von Eigenvektoren, die man orthonormal wählen kann

$$x^{(i)\dagger} x^{(j)} = \delta_{ij}, \tag{1.13}$$

so daß X_R unitär ist. Dann gilt $X_L = X_R^{-1} = X_R^\dagger$, und die Linkseigenvektoren sind die adjungierten Rechtseigenvektoren.

Noch spezieller, aber physikalisch (Quantenmechanik) von größter Bedeutung ist der hermitesche Fall, $A = A^\dagger$, woraus die Eigenschaft "normal" trivial folgt. Hier sind dann alle Eigenwerte reell, und aus Obigem folgt die unitäre Diagonalisierbarkeit. Ein Spezialfall der hermiteschen Matrizen wiederum sind die reellen symmetrischen, $A = A^T$, auf die wir uns im folgenden beschränken wollen.

1.2 Jacobi–Methode

Wir wollen nun die iterative Jacobi Methode für eine reell symmetrische Matrix A einführen. Wir wissen, daß diese orthogonal diagonalisierbar ist,

$$\mathrm{diag}(\lambda_1 \ldots \lambda_n) = O^T A O, \tag{1.14}$$

wobei die Spalten von O die Rechtseigenvektoren sind. Die Idee ist, O aus besonders einfachen orthogonalen Transformationen sukzessive aufzubauen,

$$O = P_1 P_2 P_3 \ldots, \tag{1.15}$$

d. h. man transformiert

$$A \to P_1^T A P_1 \to P_2^T P_1^T A P_1 P_2 \to \ldots \tag{1.16}$$

solange, bis die modifizierte Matrix (in der gewünschten Näherung) diagonal ist. Will man auch die Eigenvektoren, so muß man gleichzeitig die verwendeten P_k aufmultiplizieren. Bei der Wahl der P_k wird eine Strategie verfolgt, die durch die Transformationen ständig die "Fehlerfunktion"

$$S = \sum_{i \neq j} |a_{ij}|^2 \tag{1.17}$$

verkleinert. Es bleibt nun, die Form der P_k anzugeben. Hier wählt man Drehungen in einer beliebigen pq-Ebene ($p < q$) des n-dimensionalen Raumes,

$$p_{ij} = \begin{cases} 1 & \text{für } i = j \notin \{p, q\} \\ c & \text{für } i = j \in \{p, q\} \\ s & \text{für } i = p, j = q \\ -s & \text{für } i = q, j = p \\ 0 & \text{sonst} \end{cases} \tag{1.18}$$

mit $c = \cos(\phi), s = \sin(\phi)$ für den Drehwinkel ϕ. Für einen solchen Schritt $A' = P^T AP$ schreiben wir nun die Matrixelemente von A', die gegen A verändert sind ($i \notin \{p, q\}$):

$$a'_{ip} = ca_{ip} - sa_{iq} \tag{1.19}$$
$$a'_{iq} = ca_{iq} + sa_{ip} \tag{1.20}$$
$$a'_{pp} = c^2 a_{pp} + s^2 a_{qq} - 2sca_{pq} \tag{1.21}$$
$$a'_{qq} = s^2 a_{pp} + c^2 a_{qq} + 2sca_{pq} \tag{1.22}$$
$$a'_{pq} = (c^2 - s^2)a_{pq} + sc(a_{pp} - a_{qq}) \tag{1.23}$$

Verlangt man nun $a'_{pq} = 0$, so wird der Drehwinkel bestimmt durch

$$\theta = \cot(2\phi) = \frac{c^2 - s^2}{2sc} = \frac{a_{qq} - a_{pp}}{2a_{pq}}. \tag{1.24}$$

Mit $t = s/c$ führt dies auf

$$t^2 + 2t\theta - 1 = 0. \tag{1.25}$$

Es hat sich numerisch bewährt, die Lösung dieser quadratischen Gleichung zu nehmen, die zu $\phi \to 0$ führt für $a_{pq} \to 0$,

$$t = \frac{\operatorname{sgn}(\theta)}{|\theta| + \sqrt{\theta^2 + 1}}. \tag{1.26}$$

Diese Form ist gegen Rundungsfehler stabil. Sollte θ^2 den Zahlenbereich überschreiten, so kann $t = 0$ gesetzt werden. Für den Ersetzungsschritt $A \to A'$ kann man noch umschreiben

$$a'_{ip} = a_{ip} - s(a_{iq} + \tau a_{ip}) \tag{1.27}$$
$$a'_{iq} = a_{iq} + s(a_{ip} - \tau a_{iq}) \tag{1.28}$$
$$a'_{pp} = a_{pp} - ta_{pq} \tag{1.29}$$
$$a'_{qq} = a_{qq} + ta_{pq} \tag{1.30}$$
$$a'_{pq} = 0 \tag{1.31}$$

mit

$$c = \frac{1}{\sqrt{t^2+1}}, \tag{1.32}$$

$$s = tc, \tag{1.33}$$

$$\tau = \frac{s}{1+c}. \tag{1.34}$$

Durch einen solchen Schritt wird S reduziert gemäß

$$S' = S - 2|a_{pq}|^2. \tag{1.35}$$

Solange also noch Nebendiagonalelemente da sind, wird S echt kleiner, was die Konvergenz des Verfahrens beweist. Die aufeinander folgenden Schritte sind nicht unabhängig in dem Sinne, daß $a_{pq} = 0$ beim Wegtransformieren anderer Nebendiagonalelemente wieder zerstört wird. Somit handelt es sich um ein echtes Iterationsverfahren, für das in [1] quadratische Konvergenz zitiert wird. Um die Iteration durchzuführen, muß noch festgelegt werden, wie die Ebenen *pq* gewählt werden. Es ist klar, daß alle vorkommen müssen, damit alle Nebendiagonalelemente eliminiert werden können. Standard ist die zyklische Jacobi–Methode mit $pq = 12,\ 13,\ \ldots,\ 1n,\ 23{,}24{,}\ldots$.

1.3 Transformation auf Tridiagonalform

Im Verfahren des letzten Abschnitts wurden Nebendiagonalelemente von A wegtransformiert. Da diese bei den nachfolgenden Schritten nicht Null bleiben, braucht man viele Schritte um iterativ die Matrix zu diagonalisieren. Wir hatten schon gesagt, daß es keine Verfahren mit endlicher Schrittzahl geben wird. In diesem Abschnitt werden wir sehen, daß man aber A mit endlicher Schrittzahl in eine (symmetrische) Tridiagonalmatrix T transformieren kann. So entsteht

$$T = \begin{pmatrix} t_{11} & t_{12} & 0 & 0 & 0 & \cdots & 0 \\ t_{21} & t_{22} & t_{23} & 0 & 0 & \cdots & 0 \\ 0 & t_{32} & t_{33} & t_{34} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & & \cdots & 0 & t_{n-1,n-2} & t_{n-1,n-1} & t_{n-1,n} \\ 0 & & \cdots & & 0 & t_{n,n-1} & t_{nn} \end{pmatrix}, \tag{1.36}$$

wo also gilt: $t_{ij} = 0$ wenn $|i-j| > 1$ und $t_{ij} = t_{ji}$, da die Symmetrie erhalten bleibt. Diese Matrix muß dann noch immer iterativ diagonalisiert werden, wofür es aber Verfahren gibt, die wesentlich schneller konvergieren als Jacobi bei der ursprünglichen Matrix.

Zu diesem Zweck bieten sich die schon im Zusammenhang mit den linearen Gleichungssystemen eingeführten Householder Reflexionen an. Sie liefern orthogonale[1] Matrizen P_i, die, als Ähnlichkeitstransformationen angewandt, besonders effektiv zur Tridiagonalform führen. Der erste Schritt ist

$$A' = P_1 A P_1 \tag{1.37}$$

und dabei wird P_1 so konstruiert, daß $a'_{31} = \ldots = a'_{n1} = 0$ entsteht und damit wegen Symmetrie auch $a'_{13} = \ldots = a'_{1n} = 0$. Dies läßt sich erreichen mit

$$\begin{aligned}
P_1 &= 1 - \frac{uu^T}{H} && (1.38)\\
u^T &= (0, a_{21} + \sigma_1, a_{31}, \cdots, a_{n1}) && (1.39)\\
\sigma_1^2 &= \sum_{i=2}^{n} (a_{i1})^2 && (1.40)\\
\mathrm{sgn}(\sigma_1) &= \mathrm{sgn}(a_{21}) && (1.41)\\
H &= \frac{1}{2}|u|^2 && (1.42)\\
&= \sigma_1(\sigma_1 + a_{21}). && (1.43)
\end{aligned}$$

Das Element a_{11} bleibt hier ungeändert, $P_{ij} = \delta_{ij}$ wenn $i = 1$ oder $j = 1$, da sonst die Faktoren P von links und rechts miteinander auf der Diagonalen interferieren würden. Nach diesem Schritt haben wir

$$A' = \begin{pmatrix} a_{11} & -\sigma_1 & 0 & \ldots & 0 \\ -\sigma_1 & a'_{22} & a'_{23} & \cdots & \cdots \\ 0 & a'_{32} & a'_{33} & \cdots & \cdots \\ 0 & \vdots & \vdots & \vdots & \vdots \end{pmatrix}. \tag{1.44}$$

So kann man fortfahren, bis die modifizierte Matrix A Tridiagonalform hat. Im k-ten Schritt gilt

$$u^T = (0, \cdots, 0, a_{k+1,k} + \sigma_k, a_{k+2,k}, \cdots, a_{nk}) \tag{1.45}$$

[1]Wir bleiben hier bei der reellen Form, obwohl alles leicht komplex geschrieben werden kann.

$$\sigma_k^2 = \sum_{i=k+1}^{n} (a_{ik})^2 \tag{1.46}$$

$$\mathrm{sgn}(\sigma_k) = \mathrm{sgn}(a_{k+1,k}), \tag{1.47}$$

und hierbei werden die ersten $k-1$ Zeilen und Spalten sowie das aktuelle Element a_{kk} nicht verändert. Die jeweilige Modifikation von A kann man wie folgt effektiv durchführen. Nach Berechnung von σ, u, H bilden wir

$$p = \frac{Au}{H} \tag{1.48}$$

$$q = p - u\frac{u^T p}{2H} \tag{1.49}$$

$$A' = A - uq^T - qu^T. \tag{1.50}$$

Will man am Ende neben Eigenwerten auch Eigenvektoren ausgeben, so muß man das Produkt $P_1 P_2 \cdots P_{n-2}$ aufmultiplizieren und als orthogonale Matrix speichern.

Das folgende Demoprogramm in MATLAB führt die Formeln vor:

```
%
% file house.m
%
% demoprogramm fuer Householder Reduktion
%
n=5;
disp('Ausgangsmatrix:');
a=rand(n,n);                % Zufallsmatrix
a = a + a.'                 % symmetrisch
%
O=eye(n);                   % Einheitsmatrix
% Reduktionsloop:
for k=1:n-2,
  fprintf('Schritt: %i',k)
  sigma=norm(a(k+1:n,k));
  if a(k+1,k) < 0, sigma = -sigma; end;       %Vorzeichen fixiert
  u = [zeros(k,1) ; a(k+1:n,k)]; u(k+1)=u(k+1)+sigma;
  H = 0.5*norm(u)^2;
  p = a*u/H;
```

```
  q = p - u*((u.'*p)/(2*H));
  O = O -O*u*u.'/H;       % Transformation
  a = a - u*q.' - q*u.'  % modifiziertes A
end
disp('Transformation:');
O
```

Sein Ouput ist

```
Ausgangsmatrix:

a =

    1.9003    0.9932    1.2223    0.8917    0.9492
    0.9932    0.9129    0.8104    1.7569    0.7976
    1.2223    0.8104    1.8436    1.6551    0.9894
    0.8917    1.7569    1.6551    0.8205    0.9035
    0.9492    0.7976    0.9894    0.9035    0.2778

Schritt: 1
a =

    1.9003   -2.0437    0.0000    0.0000    0.0000
   -2.0437    4.4473   -0.0570   -0.7092    0.5329
    0.0000   -0.0570    0.5728   -0.0098    0.1672
    0.0000   -0.7092   -0.0098   -0.9323   -0.2693
    0.0000    0.5329    0.1672   -0.2693   -0.2329

Schritt: 2
a =

    1.9003   -2.0437    0.0000    0.0000    0.0000
   -2.0437    4.4473    0.8889         0    0.0000
    0.0000    0.8889   -0.4310    0.2064    0.3475
    0.0000    0.0000    0.2064   -0.0440   -0.6799
    0.0000    0.0000    0.3475   -0.6799   -0.1174

Schritt: 3
a =
```

```
   1.9003   -2.0437    0.0000    0.0000    0.0000
  -2.0437    4.4473    0.8889    0.0000    0.0000
   0.0000    0.8889   -0.4310   -0.4041    0.0000
   0.0000    0.0000   -0.4041   -0.6954   -0.2929
   0.0000    0.0000    0.0000   -0.2929    0.5340

Transformation:

O =

   1.0000         0         0         0         0
        0   -0.4860    0.1080    0.6938   -0.5204
        0   -0.5981   -0.0207    0.1713    0.7826
        0   -0.4363   -0.7661   -0.3880   -0.2688
        0   -0.4645    0.6333   -0.5821   -0.2108
```

1.4 Eigenwerte von tridiagonalen Matrizen, QL–Methode

Die betrachtete reelle symmetrische Matrix A können wir zerlegen

$$A = QL, \tag{1.51}$$

wobei Q orthogonal ist und L eine untere Dreiecksmatrix. Dies ist wieder möglich durch Householder–Reflexionen[2]. Nun bildet man das modifizierte A' als

$$A' = LQ = Q^T A Q, \tag{1.52}$$

das also zu A orthogonal ähnlich ist und dieselben Eigenwerte hat. Es gibt einen Satz [1], der besagt, daß die Iteration dieses Schrittes eine beliebige reelle Matrix in Dreiecksform bringt mit den Eigenwerten auf der Diagonalen[3]. Direkt auf das voll besetzte A angewandt ist diese Methode mit $\mathrm{O}(n^3)$ Operationen pro Iteration aber nicht effektiv, wohl aber für tridiagonale, wo man mit $\mathrm{O}(n)$ auskommen kann. Dabei ist wichtig, daß sich auch noch zeigen läßt, daß A' wieder tridiagonal ist, wenn dies für A gilt. Dann werden die garantierten Nullen natürlich nicht berechnet. Im tridiagonalen Fall ist

[2]Hier muß man zuerst führende Nullen in der letzten Spalte von A erzeugen etc.

[3]Im Fall entarteter $|\lambda_i|$ gibt es für volle Matrizen noch Komplikationen die für tridiagonale aber keine Rolle spielen.

es Standard, Q aus Rotationen wie beim Jacobi–Verfahren aufzubauen. Man wählt in jeder Iteration die Folge von Ebenen 12, 23, . . . , $n-1n$, um die Elemente $a_{12}, a_{23}, \ldots, a_{n-1n}$ zu annihilieren. Damit wird die tridiagonale Matrix symmetrisch dreieckig, also diagonal. Aus der hierbei entstandenen Ähnlichkeitstransformation, zusammen mit der vom Householder–Schritt, kann man auch die Eigenvektoren gewinnen.

Über die Konvergenz wird in [1] gesagt, daß im s-ten Schritt in der Matrix $A^{(s)}$ die Elemente oberhalb der Diagonalen gegen Null gehen wie

$$a_{ij}^{(s)} \propto \left(\frac{\lambda_i}{\lambda_j}\right)^s, \tag{1.53}$$

wobei die Eigenwerte der Größe nach geordnet sind, so daß $|\lambda_i| < |\lambda_j|$ gilt[4]. Man kann diese Konvergenz u. U. verbessern durch eine Verschiebung der Eigenwerte $A \to A + k1$.

1.5 Eigenwertbestimmung in MATLAB

Es gibt in MATLAB wieder ein fertiges Programm mit Namen **eig** zur Bestimmung von Eigenwerten und Eigenvektoren:

```
>> help eig
 EIG    Eigenvalues and eigenvectors.
    E = EIG(X) is a vector containing the eigenvalues of a square
    matrix X.

    [V,D] = EIG(X) produces a diagonal matrix D of eigenvalues and a
    full matrix V whose columns are the corresponding eigenvectors so
    that X*V = V*D.

    [V,D] = EIG(X,'nobalance') performs the computation with balancing
    disabled, which sometimes gives more accurate results for certain
    problems with unusual scaling. If X is symmetric, EIG(X,'nobalance')
    is ignored since X is already balanced.

    E = EIG(A,B) is a vector containing the generalized eigenvalues
    of square matrices A and B.
```

[4]Für $\lambda_i = \lambda_j$ läßt sich zeigen, daß $a_{ij} = 0$ schon nach dem Tridiagonalisieren gilt.

```
    [V,D] = EIG(A,B) produces a diagonal matrix D of generalized
    eigenvalues and a full matrix V whose columns are the
    corresponding eigenvectors so that A*V = B*V*D.

    EIG(A,B,'chol') is the same as EIG(A,B) for symmetric A and symmetric
    positive definite B.  It computes the generalized eigenvalues of A and B
    using the Cholesky factorization of B.
    EIG(A,B,'qz') ignores the symmetry of A and B and uses the QZ algorithm.
    In general, the two algorithms return the same result, however using the
    QZ algorithm may be more stable for certain problems.
    The flag is ignored when A and B are not symmetric.
```

Mit der im vorletzten Abschnitt erzeugten Matrix a erhält man:

```
>> [v,d]=eig(a)

v =

   -0.4716    0.0930   -0.1934    0.8505   -0.0910
    0.8725    0.1362   -0.2121    0.4145   -0.0577
    0.1274   -0.6353    0.6976    0.3031    0.0404
   -0.0081   -0.7425   -0.6225   -0.0896   -0.2304
    0.0005   -0.1336   -0.2085    0.0709    0.9663


d =

    5.6816         0         0         0         0
         0   -1.0939         0         0         0
         0         0   -0.3406         0         0
         0         0         0    0.9042         0
         0         0         0         0    0.6038

>> for k=1:n, delta(k)=norm(a*v(:,k) - d(k,k)*v(:,k)); end
>> delta

delta =
```

```
1.0e-14 *

  0.5348    0.0750    0.0331    0.0920    0.0356
```

1.6 Singular Value Decomposition

Die Singular Value Decomposition (SVD) ist ein Verfahren zur Lösung des linearen Gleichungssystems $Ax = b$. Es ist von Interesse, wenn es lineare Abhängigkeiten zwischen Zeilen oder Spalten gibt, also für $m \times n$–Matrizen A mit $m \neq n$ oder für quadratische singuläre Probleme. Für letzteren Fall wollen wir uns hier interessieren. Eigentlich gehört dieser Abschnitt ins Kapitel über lineare Gleichungssysteme, aber weil das Verfahren wesentlich auf der Bestimmung eines Eigenwertspektrums beruht, wird es hier besprochen.

In der SVD hat man

$$A = UWV^T \tag{1.54}$$

wobei U und V orthogonal sind und W diagonal ist mit Diagonalelementen $w_i \geq 0$. Mit Hilfe der Spaltenvektoren $u^{(i)}$, $v^{(i)}$ von U bzw. V gilt auch

$$A = \sum_{i=1}^{n} u^{(i)} w_i \, v^{(i)T}. \tag{1.55}$$

Mathematische Sätze [1] garantieren, daß es diese Zerlegung immer gibt, auch im singulären Fall. Sie ist nahezu eindeutig. Sind alle w_i verschieden, so bleibt nur ein gemeinsamer Vorzeichenwechsel bei $u^{(i)}$, $v^{(i)}$ und gemeinsame Permutationen von $u^{(i)}$, $v^{(i)}$ und w_i, über die man durch Ordnen der w_i verfügen kann. Sind mehrere w_i gleich, so kann man in den zugehörigen Unterräumen die Basen der $u^{(i)}$, $v^{(i)}$ rotieren.

Für den Fall, daß A regulär ist, sind alle $w_i > 0$ und das Inverse ist gegeben durch

$$A^{-1} = VW^{-1}U^T \tag{1.56}$$

mit trivialem W^{-1}. Im singulären Fall sei angenommen, es verschwinden[5] die letzten k Diagonalelemente $w_n, w_{n-1}, \ldots w_{n-k+1}$. Dann sind die zugehörigen Vektoren $v^{(i)}$ eine Basis des Nullraumes der durch A gegebenen linearen Abbildung. Die ersten $n - k$ Vektoren $u^{(i)}$ bilden eine Basis des vom Bild von A aufgespannten Raumes. Das lineare System $Ax = b$ hat offenbar nur eine

[5] In der Praxis wird man hier wegen Rundungsfehlern eine numerische Schranke setzen.

Lösung, wenn b im Bild von A liegt, und diese ist dann um beliebige Vektoren aus dem Nullraum unbestimmt.

Die Durchführung der SVD ist kompliziert und soll hier nicht dargestellt werden. Programme in Fortran, C und weiteren Sprachen gibt es in [1], und auch MATLAB stellt eine (binär implementierte) äusserst stabile und bewährte Routine **svd** bereit:

```
>> A=rand(4,4);
>> A(1,:)=A(2,:)*pi

A =

    0.6243    0.0480    1.4640    0.6366
    0.1987    0.0153    0.4660    0.2026
    0.6038    0.7468    0.4186    0.6721
    0.2722    0.4451    0.8462    0.8381

>> [U,W,V] = svd(A)

U =

   -0.6876    0.5911   -0.2928    0.3033
   -0.2189    0.1882   -0.0932   -0.9529
   -0.4463   -0.7562   -0.4786   -0.0000
   -0.5292   -0.2083    0.8225    0.0000


W =

    2.3861         0         0         0
         0    0.8291         0         0
         0         0    0.3130         0
         0         0         0    0.0000


V =

   -0.3714   -0.1288   -0.8512    0.3476
```

```
   -0.2536   -0.7553   -0.0217   -0.6040
   -0.7306    0.5552    0.0755   -0.3901
   -0.5137   -0.3236    0.5189    0.6018

>> for k=1:4, delta(k)=norm(A*V(:,k) - W(k,k)*U(:,k)); end
>> delta

delta =

   1.0e-14 *

    0.0801    0.1325    0.0312    0.0620
```

2 Fourier Transformation

Die Fourier Analyse ist von großer Bedeutung in der Physik und darüber hinaus. In CP1 sind wir z. B. bei der Elektrostatik (Konvergenzanalyse der Jacobi Relaxation) damit in Berührung gekommen, und zwar speziell für ein endliches diskretes Gitter. Wir wollen daher nun diese Version der Fourieranalyse und auch die übrigen Verwandten (Fourier Integral, unendliche Reihen) systematisch zusammenstellen und diskutieren. Es wird alles hergeleitet, allerdings ohne mathematische Strenge, z. B. werden Konvergenzfragen ausgeklammert.

Warum sind Fourier Argumente und ebene Wellen in der Physik derart allgegenwärtig? Der Grund ist wohl darin zu sehen, dass die meisten physikalischen Gesetze aus Differenzialgleichungen bestehen, worin Ableitungen wie dx/dt vorkommen. Kann man nun $x(t)$ entwickeln

$$x(t) = \sum_{\omega_n} \alpha_n \mathrm{e}^{i\omega_n t}, \tag{2.1}$$

so gilt bekanntlich

$$\frac{dx}{dt} = \sum_{\omega_n} i\omega_n \alpha_n \mathrm{e}^{i\omega_n t}. \tag{2.2}$$

Aus der komplizierten Operation $x(t) \to dx/dt$ werden auf dem Niveau der Fourier Amplituden einfache Multiplikationen $\{\alpha_n\} \to \{i\omega_n\alpha_n\}$, was man z.B. auch formal invertieren kann (solange $\omega_n \neq 0$ für alle n). Es ist natürlich zu diskutieren, welche ω_n vorkommen etc.

Später kommen wir schließlich auf das FFT (Fast Fourier Transform) Verfahren, einen besonders effektiven numerischen Algorithmus um Fourier Transformation durchzuführen, der bei Bildverabeitung und anderen Gebieten ein große Rolle spielt. Hier wird ein Kostenfaktor, der zunächst proportional zu N^2 steigt (N: Anzahl der Variablen bzw. Freiheitsgrade) reduziert auf $N \ln N$, im wesentlichen also um einen Faktor N.

2.1 Endliches Gitter

Wir betrachten zuerst einen diskretisierten endlichen Abschnitt der x-Achse bestehend aus den Punkten $\{x\} = \{0, a, 2a, \ldots, L-a\}$. Er reicht also vom Ursprung bis $L-a$ und hat Punkte mit Abstand a. Bis auf Weiteres meint $\sum_x \ldots$ immer Summen über diese $N = L/a$ Elemente. Darauf betrachten wir komplexe Funktionen $f(x)$. Das ist natürlich nur eine besondere (später hilfreiche) Indizierung der Komponenten eines Vektors in $\mathbb{C}^N$. Eine entscheidende Formel ist nun

$$\frac{a}{L}\sum_k \mathrm{e}^{ik(x-y)} = \delta_{xy} \tag{2.3}$$

mit dem Kronecker δ rechts und der $k-$ Summe über die Menge $\{k\} = \{0, 2\pi/L, 4\pi/L, 6\pi/L, \ldots, (L/a-1)2\pi/L\}$ (auch N Elemente). Der Beweis benötigt nur die Summenformel für eine endliche geometrische Reihe. Wir setzen dazu $x = la, y = ma, k = n2\pi/L, 0 \leq l, m, n < N$. Dann wird aus (2.3)

$$\frac{1}{N}\sum_{n=0}^{N-1}\left[\mathrm{e}^{i2\pi(l-m)/N}\right]^n = \begin{cases} 1 & \text{wenn} \quad l = m \\ \frac{1}{N}\frac{1-\mathrm{e}^{i2\pi(l-m)}}{1-\mathrm{e}^{i2\pi(l-m)/N}} = 0 & \text{wenn} \quad l-m \neq 0, \end{cases} \tag{2.4}$$

was mit (2.3) zusammenfällt. Man beachte, dass im unteren Fall der Nenner nicht verschwindet. Mit dieser Formel können wir trivial auf einen Schlag die Fourier Hin-und Rücktransformation bekommen

$$f(x) = \sum_y \delta_{xy} f(y) = \frac{1}{L}\sum_k \mathrm{e}^{ikx}\tilde{f}(k), \tag{2.5}$$

was zur Identität wird mit der Festlegung

$$\tilde{f}(k) = a\sum_y \mathrm{e}^{-iky} f(y). \tag{2.6}$$

Bemerkungen dazu:

- Da es bisher nur endliche Summen gibt, sind keine Subtilitäten wie Konvergenz zu diskutieren.
- Hier sind eine Reihe von Konventionen getroffen, die viele aber nicht alle Leute teilen. Die Vorfaktoren können anders auf Hin- und Rücktransformation verteilt werden. Vorzeichen im Exponenten können anders gewählt werden (es kommt aber stets einmal $+$ und einmal $-$vor).
- $f, \tilde{f}$ sind komplex. Das ist der einfachste Fall, reelle Spezialisierungen, s. unten.
- Die Größen haben (konsistente) Dimensionen wenn x bzw. a Längen sind, bitte verifizieren.

- Man kann statt Funktionen vom Ort auch z. B. die Zeit wählen (dann t statt x, ω statt k), der Mathematik ist das egal.

In (2.5) wird $f(x)$ als Überlagerung von 'Funktionen' e^{ikx} dargestellten. Für die verwendeten Werte k sind diese periodisch in x mit Periode L. Daher ist f durch die rechte Seite für alle (diskretisierten) $x = la, l \in \mathbb{Z}$ auf der x-Achse definiert, wiederholt sich aber,

$$f(x \pm L) = f(x), \tag{2.7}$$

es gibt also weiter nur N unabhängige Werte. Es ist daher vernünftig sich f von vornherein so erweitert, d. h. periodisch fortgesetzt, vorzustellen. Diese Art, ein System endlich zu machen wird in der Physik auch als periodische Randbedingungen bezeichnet und spielt eine große Rolle. Nun ist der Summand auf der rechten Seite von (2.6) periodisch in y und man kann die Summe über y über einen beliebigen Abschnitt von N aufeinanderfolgenden (diskreten) Werten auf der x-Achse führen mit immer dem gleichen Resultat. Ein Beispiel wäre für ungerades $N = 2M + 1$ die manifest symmetrische Wahl $y = -Ma, -(M-1)a, \ldots, -a, 0, a, \ldots, Ma$. Der Faktor e^{ikx} ist für jedes (diskrete) x auch in k periodisch mit der Periode $2\pi/a$. Damit gilt entsprechendes für $\tilde{f}(k)$ und die k-Summen. Die Periodizität der Wellenzahlen k ist also durch die Diskretisierung im Ort bestimmt. Auch das in (2.3) dargestellte δ_{xy} ist somit periodisch fortgesetzt, z.B. $\delta_{xy} = \delta_{x(y\pm L)}$. Aus dem obigen ist nun leicht zu zeigen, dass dual zu (2.3) gilt

$$\frac{a}{L} \sum_x \mathrm{e}^{ix(k-k')} = \delta_{kk'}. \tag{2.8}$$

2.2 Beispiel: Helmholtz

Eine eindimensionale diskretisierte Helmholtz Gleichung[6] ist zu lösen

$$(-\Delta + \mu^2) f(x) = \rho(x). \tag{2.9}$$

Hier ist Δ die schon diskutierte (z. B. Elektrostatik) Standarddiskretisierung der zweiten Ableitung

$$-\Delta f(x) = 2f(x) - f(x+a) - f(x-a), \tag{2.10}$$

[6]Die Poissongleichung ergibt sich für $\mu \to 0$. Dann muss man aber die $k = 0$ Mode physikalisch anders fixieren, sonst wird der Operator singulär.

und wir setzen periodische Randbedingungen mit Periode L voraus. Es sei nun jedem überlassen zu zeigen, dass für die zugehörigen Fourier Amplituden gilt

$$(\hat{k}^2 + \mu^2)\tilde{f}(k) = \tilde{\rho}(k), \quad \hat{k} = \frac{2}{a}\sin(ka/2). \tag{2.11}$$

Wie die Notation suggeriert wird $\hat{k} \approx k$ für Moden mit $ka \ll 1$. Das sind genau diejenigen, für die die Diskretisierung eine gute Näherung des Kontinuumsverhalten liefert. Wenn ρ und damit f hohe Moden der Größenordnung $1/a$ enthält, gilt das nicht, die Werte schwanken nennenswert von einem Gitterplatz zum nächsten, und alle Resultate sind diskretisierungsabhängig. Wenn das physikalische Problem kontinuierlich ist, spricht man dann von Artefakten. Die Wellenzahlen sind durch das Gitter nach oben begrenzt, da offenbar $|\hat{k}| \leq 2/a$ gilt (Kurzdistanz oder Ultraviolett Regularisierung).

Man kann nun auch den zu $-\Delta+\mu^2$ inversen Operator als Matrix angeben

$$f(x) = a\sum_y K(x,y)\rho(y), \tag{2.12}$$

$$K(x,y) = \frac{1}{L}\sum_k \frac{\mathrm{e}^{ik(x-y)}}{\hat{k}^2 + \mu^2}, \tag{2.13}$$

der Hin- und Rücktransformation enthält mit der Divison dazwischen. Sie entspricht der Inversion des im 'k-Raum' diagonalen Operators. Dies ist eine extrem wichtige Anwendung, u. a. in der Quantenfeldtheorie. Die mehrdimensionale Verallgemeinerung (s. unten) ist kein Problem.

2.3 Kontinuumslimes

Man kann verschiedene Limites der obigen Formeln betrachten, hier zunächst $a \to 0$ bei festem L. Offenbar wird dann die Anzahl der Wellenzahlwerte L/a divergieren und man bekommt es im Analogon zu (2.5) mit unendlichen Summen und in (2.6) mit Integralen als Grenzfall einer Riemann Summe zu tun. Hier muss man nun bei der oben diskutierten freien Wahl der Summationsperioden Pathologien vermeiden. Wir wollen das hier plausibel machen, können es aber nicht streng diskutieren. Wir betrachten als Beispiel die langsam veränderliche periodische Funktion $f(x) = \exp(-i2\pi x/L)$ beliebig fein diskretisiert. Wenn wir nun den (endlichen) Satz der $\{k\}$ wie oben positiv wählen, so hat diese Funktion nur eine Fourierkomponente bei $k = (2\pi/L)(L/a - 1)$, dem größten k. Um einen nichtpathologischen Grenzübergang zu erhalten, sollte aber eine so glatte Funktionen in kleine $|k|$ übergehen. Das wird gewährleistet, wenn wir schon vor dem Grenzübergang symmetrisch $-\pi/a < k \leq \pi/a$ wählen, was ab sofort in allen Summen gelten soll. Absolute Symmetrie mit Null in der Mitte ist dann offenbar nur für ungerade N möglich und wir wollen uns darauf beschänken[7]. Dann gehen (2.6),(2.5),(2.3) über in

$$f(x) = \frac{1}{L}\sum_k \mathrm{e}^{ikx}\tilde{f}(k), \quad k = 0, \pm\frac{2\pi}{L}, \pm\frac{4\pi}{L}, , \pm\frac{6\pi}{L}, \ldots (\infty \text{ viele Terme}) \tag{2.14}$$

$$\tilde{f}(k) = \int_0^L dy \mathrm{e}^{-iky} f(y) = \int_{-L/2}^{L/2} dy \mathrm{e}^{-iky} f(y), \tag{2.15}$$

$$\frac{1}{L}\sum_k \mathrm{e}^{ik(x-y)} = \frac{1}{a}\delta_{xy} \to \delta(x-y). \tag{2.16}$$

Wir haben nun im Raum L-periodische Funktionen (gilt auch für $\delta(.)$) und unendliche Summen über Wellenzahlen, deren Konvergenz für die entwickelten Funktionen vorausgesetzt sei. In dieser Form — unendliche Fourier Reihen für periodische Funktionen — tritt das Thema meist zuerst auf in der Mathe Ausbildung.

2.4 Unendlicher Volumenlimes

Man kann auch $L \to \infty$ bei endlichem a nehmen. Dann wird $\tilde{f}$ zur Funktion auf dem endlichen Intervall $(-\pi/a, \pi/a)$, der Brillouin Zone, und die Summationen über Punkte auf der x-Achse unendlich (unendliches Gitter). Dies ist ein wichtiger Fall in der Festkörperphysik. Unsere Formeln lauten nun

$$f(x) = \int_{-\pi/a}^{\pi/a} \frac{dk}{2\pi} \mathrm{e}^{ikx}\tilde{f}(k) \tag{2.17}$$

und $\tilde{f}(k)$ ist wie in (2.6) gegeben. Der Integrand hat wieder die Periode $2\pi/a$, und der Integrationsbereich kann daher auch anders gelegt werden. Die umgekehrte Transformation bleibt wie in (2.6), nur dass die Summe nun

[7] Tatsächlich würde die geringe scheinbare Asymmetrie bei geradem N keine Rolle spielen, nur die Formeln sind hässlicher.

über $y = am, m = -\infty, \ldots, \infty$ geht. In (2.17) gab es den Übergang von Summe zu Integral

$$\frac{1}{L} \sum_{k=\frac{2\pi}{L}n} \ldots = \frac{1}{2\pi} \sum_{k=\frac{2\pi}{L}n} \frac{2\pi}{L} \ldots . \longrightarrow \frac{1}{2\pi} \int dk \ldots .$$

2.5 Fourierintegral

Nun wollen wir den doppelten Limes $a \to 0$ und $L \to \infty$ nehmen. Vor dem Grenzübergang sind die x- und k-Bereiche symmetrisch zu Null zu wählen. Es resultieren die Standardformeln der Fourier Integration

$$f(x) = \int_{-\infty}^{\infty} \frac{dk}{2\pi} \mathrm{e}^{ikx} \tilde{f}(k), \tag{2.18}$$

$$\tilde{f}(k) = \int_{-\infty}^{\infty} dx \mathrm{e}^{-ikx} f(x), \tag{2.19}$$

die z. B. in der Quantenmechanik ($k = p/\hbar$) wichtig sind. Die Masterformel lautet

$$\int_{-\infty}^{\infty} \frac{dk}{2\pi} \mathrm{e}^{ikx} = \delta(x). \tag{2.20}$$

Hier muss man sich erinnern, dass wir es nun mit Distributionen oder verallgemeinerten Funktionen zu tun haben. D.h. lax gesprochen, dass obige Identität in Integralen mit weiteren genügend 'braven' (glatt, abfallend) Funktionen zu verwenden ist. Nur so ist ja die δ-Funktion definiert.

Es gibt übrigens noch eine elegante Art, ohne die diskreten Formeln direkt zu (2.20) zu gelangen. Mit einem zusätzlichen Konvergenzfaktor haben wir

$$\int_{-\infty}^{\infty} \frac{dk}{2\pi} \mathrm{e}^{ikx - k^2\varepsilon^2/2} = \frac{1}{\varepsilon\sqrt{2\pi}} \mathrm{e}^{-x^2/(2\varepsilon^2)} = \delta_\varepsilon(x). \tag{2.21}$$

Hier haben wir durch Ausführen des Gauss-Integrals eine regularisierte Darstellung von δ bekommen mit der Breite ε (und Integral Eins). Die verallgemeinerten Funktionen in (2.20) entstehen auf beiden Seiten als Grenzfälle dieser gewöhnlichen Funktionen von x für $\varepsilon \to 0$.

2.6 Abtasttheorem

Hier verwenden wir die $\omega - t$ Sprache. Gegeben sei ein kontinuierliches Audio Signal $f(t)$, das wir aufzeichnen wollen. Seine Fourier Transformierte ist

$$\tilde{f}(\omega) = \int_{-\infty}^{\infty} dt \mathrm{e}^{-i\omega t} f(t). \tag{2.22}$$

Wir wissen, dass jede Schallquelle nur ein endliches Spektrum hat und postulieren

$$\tilde{f}(\omega) = 0 \text{ für } |\omega| > \omega_{\max}.$$

Nun wissen wir aber, dass die Information von $\tilde{f}$ auf einem kompakten Intervall einer diskretisierten Funktion in x bzw. nun in t entspricht. Setzen wir $\omega_{\max} = \pi/\tau$, so können wir das gleiche $\tilde{f}(\omega)$, und damit die gesamte Information, gewinnen, wenn wir nur $f_n = f(n\tau), n \in Z$ kennen. Also ist $f(t)$ durch dieses diskrete sampling vollständig bestimmt. Man spricht von der kritischen Nyquist Frequenz

$$f_{\mathrm{Ny}} = \frac{\omega_{\max}}{2\pi} = \frac{1}{2\tau}, \tag{2.23}$$

die Abtastrate $1/\tau$ ist gleich der *doppelten* Nyquist Frequenz.

Die Rekonstruktion des Signals (exakte Interpolation) funktioniert mit diskreter Hin- und kontinuierlicher Rücktransformation

$$f(t) = \int_{-\pi/\tau}^{\pi/\tau} \frac{d\omega}{2\pi} \mathrm{e}^{i\omega t} \tau \sum_{n=-\infty}^{n=\infty} \mathrm{e}^{-i\omega\tau n} f(\tau n) = \frac{\tau}{\pi} \sum_{n=-\infty}^{n=\infty} \frac{\sin[2\pi f_{\mathrm{Ny}}(t-\tau n)]}{t-\tau n} f_n. \tag{2.24}$$

Wählt man τ zu groß, so wird nicht über das vollständige Spektrum integriert. Gleichzeitig enthalten die bei sampling gewonnen Werte Beiträge der höheren Spektralanteile, die sozusagen nach unten transferiert werden. Dieser Effekt heisst aliasing und verfälscht natürlich das Signal [1].

2.7 Mehrere Dimensionen

Es ist einfach von der einen Dimension x hier auf mehrere, z.B. $\vec{x}$ zu verallgemeinern. Diskret z.B.

$$f(\vec{x}) = \frac{1}{L^3} \sum_{\vec{k}} \mathrm{e}^{i\vec{k}\vec{x}} \tilde{f}(\vec{k}) \tag{2.25}$$

und

$$\tilde{f}(\vec{k}) = a^3 \sum_{\vec{x}} \mathrm{e}^{-i\vec{k}\vec{x}} f(\vec{x}). \qquad (2.26)$$

Hier ist in jeder Komponente von $\vec{x}$ gleich diskretisiert worden und jede Komponente von $\vec{k}$ läuft unabhängig über die gleichen Werte wie vorher. Ein triviale Verallgemeinerung wäre es, die relevanten Skalen verschieden zu wählen, $a_i, L_i, i = 1, 2, 3$.

Die verschiedenen Limites sehen analog aus. Die Integral Masterformel wird z.B.

$$\int_{-\infty}^{\infty} \frac{d^3k}{(2\pi)^3} \mathrm{e}^{i\vec{k}\vec{x}} = \delta^3(\vec{x}). \qquad (2.27)$$

2.8 Algorithmische Implementierung, FFT

In vielen Fällen werden die Transformationen numerisch durchgeführt. Dabei beschränken wir uns auf die diskrete Form. Oft ist es günstiger, ein physikalisches System mit plausiblen Skalen zu diskretisieren oder in ein endliches Volumen zu setzen, als Fourier Integrale — assoziiert mit unendlich vielen Freiheitsgraden — numerisch mit Simpson o. ä. durchzuführen.

Klarerweise können wir in (2.5) und (2.6) Matrixmultiplikationen erblicken mit den $N \times N$ Matrizen mit Elementen $\exp(\pm ipx)$, was bekanntlich $\sim N^2$ Multipikationen kostet für Matrix $\times$ Vektor.

Im mehrdimensionalen Fall, z.B. 3-dimensional wie in (2.25) könnte man naiv meinen, dass daraus nun $(N^3)^2$ wird, was glücklicherweise nicht der Fall ist. Man kann so vorgehen, dass man zunächst transformiert

$$\tilde{f}(k_1, k_2, k_3) \to f^{(1)}(x_1, k_2, k_3) = \frac{1}{L} \sum_{k_1} \mathrm{e}^{ik_1x_1} \tilde{f}(k_1, k_2, k_3), \qquad (2.28)$$

also nur in der ersten Variablen transformiert. Da man dies für jeden Wert der "Zuschauervariablen" k_2, k_3 einmal tun muss, sind die Kosten N^4. Offenbar ist man mit zwei weiteren solchen Schritten am Ziel. Die Kosten einer D-dimensionalen Transformation sind also $D \times N^{D+1}$ und nicht N^{2D}.

Der gefeierte und wichtige Algorithmus der Fast Fourier Transformation[8] [1] — nun wieder eindimensional — benutzt eine ähnliche Strategie, um die

[8]Das Verfahren wird oft Cooley und Tuckey zugeschrieben. Es soll aber schon vorher in anderen Computercodes ohne viel Aufhebens implementiert gewesen sein. Laut [1] hat auch Gauss 1805 etwas ähnliches gehabt um Flugbahnen von Asteroiden zu analysieren.

Kosten von N^2 auf $\sim N \ln N$ zu reduzieren. Wenn N nicht eine Zweierpotenz ist, wird die Sache technisch kompliziert und wir beschränken uns ab hier auf den Fall

$$N = \frac{L}{a} = 2^\nu .$$

Mit ein paar Abkürzungen können wir (2.5) in die Form bringen

$$f(an) = F(n) = \sum_{m=0}^{N-1} z^{nm} \tilde{F}(m), \quad \tilde{F}(m) = \frac{1}{L} \tilde{f}(m2\pi/L), \quad z = \exp(2\pi i/N). \tag{2.29}$$

Wegen $z^N = 1$ können wir die Multiplikation nm im Exponenten trunkieren und den Überlauf jenseits $2^\nu - 1$ weglassen. Wir schreiben n in dualer Darstellung mit ν Ziffern $n_i = 0, 1$

$$n = \sum_{i=0}^{\nu-1} n_i 2^i, \quad F(n) \equiv F(n_0, n_1, \ldots, n_{\nu-1}). \tag{2.30}$$

Für $m, \tilde{F}$ ist die Bit-gespiegelte Version technisch besser,

$$m = \sum_{i=0}^{\nu-1} m_i 2^{\nu-1-i}, \quad \tilde{F}(m) \equiv \tilde{F}(m_0, m_1, \ldots, m_{\nu-1}). \tag{2.31}$$

Das Produkt lässt sich schreiben

$$nm|_{\text{trunk}} = m_0 2^{\nu-1} n_0 + m_1 2^{\nu-2}(n_0 + n_1 2) + m_2 2^{\nu-3}(n_0 + n_1 2 + n_2 2^2) + \ldots \tag{2.32}$$

Nun kann man ähnlich vorgehen wie in (2.28):

$$\begin{aligned}
\tilde{F}(m_0, m_1, \ldots, m_{\nu-1}) & \quad \rightarrow \\
F^{(1)}(n_0, m_1, \ldots, m_{\nu-1}) &= \sum_{m_0=0,1} z^{m_0 2^{\nu-1} n_0} \tilde{F}(m_0, m_1, \ldots, m_{\nu-1}) \rightarrow \\
F^{(2)}(n_0, n_1, m_2 \ldots, m_{\nu-1}) &= \sum_{m_1=0,1} z^{m_1 2^{\nu-2}(n_0+n_1 2)} F^{(1)}(n_0, m_1, \ldots, m_{\nu-1}) \rightarrow \ldots \\
F^{(\nu)}(n_0, n_1, n_2 \ldots, n_{\nu-1}) &= F(n_0, n_1, \ldots, n_{\nu-1}).
\end{aligned} \tag{2.33}$$

Das Ganze faktorisiert zwar nicht wie im mehrdimensionalen Fall, aber ähnlich wie bei backsubstitution oder LU Zerlegung gibt es einen ‘dreieckigen’

Zusammenhang und man kann in der geeigneten Reihenfolge alles auflösen. Aufwand: Neben einigem overhead am Anfang muss man $\nu = \mathrm{ld}(N)$ mal eine Multiplikation durchführen mit $2^{\nu-1} = N/2$ Werten der "Zuschauervariablen", insgesamt[9] $\sim N\mathrm{ld}(N)$ Operationen. In praktischen Realisierungen, was wir hier nicht weiter diskutieren wollen, bringt man zuerst $\tilde{F}(m)$ in die Bit-gespiegelte Anordnung im Speicher. Die Vorfaktoren (z-Potenzen) können durch Rekursion berechnet werden, und man benötigt nicht viele trigonometrische Funktionsaufrufe.

2.9 Reelle Funktionen

Oft haben wir in der Physik reelle $f(x)$, was natürlich eine Halbierung der Anzahl der (reellen) Komponenten in Orts- und Wellenzahlraum bedeutet. Die Einschränkung ist für die verschiedenen Varianten der Fourier Entwicklung analog. In jedem Fall gilt

$$f(x)\text{reell} \Leftrightarrow \tilde{f}^*(k) = \tilde{f}(-k). \tag{2.34}$$

Hat man in den Fällen mit $a > 0$ die k-Summe (oder das Integral) nicht symmetrisch gewählt, so muss man die Periodizität von $\tilde{f}$ verwenden um $-k$ in diesen Bereich zurückzubringen, z. B. $\tilde{f}(k)^* = \tilde{f}(\frac{2\pi}{a} - k)$. Die manifest reelle Entwicklung bekommen wir z. B. aus (2.5) zunächst in der Form

$$\begin{aligned} f(x) &= \frac{1}{2L}\sum_k \left(\mathrm{e}^{ikx}\tilde{f}(k) + \mathrm{e}^{-ikx}\tilde{f}(-k)\right) \\ &= \frac{1}{L}\sum_k \left(\cos(kx)\mathrm{Re}(\tilde{f}(k)) - \sin(kx)\mathrm{Im}(\tilde{f}(k))\right). \end{aligned} \tag{2.35}$$

Soweit wird noch über alle N k-Werte summiert. Da aber nun der Summand symmetrisch in k ist kommen viele Terme (nicht alle) doppelt vor. Man kann daher schreiben

$$f(x) = \sum_{0\le k\le \pi/a} \alpha(k)\cos(kx) + \sum_{0<k<\pi/a} \beta(k)\sin(kx). \tag{2.36}$$

Bitte verifizieren Sie, wie sich α, β durch $\tilde{f}$ ausdrücken, und dass es immer N reelle Parameter gibt (N gerade, ungerade unterscheiden).

[9] ld steht für den dualen Logarithmus, Basis 2. Eigentlich ist die Basis natürlich hier egal, da sie nur den Vorfaktor betrifft.

2.10 Eingeklemmte Fourier Entwicklung

Häufig hat man es in der Physik mit Funktionen zu tun, die am Rand selbst oder deren Ableitungen verschwinden wegen Randbedingungen. Z.B. eingespannte Saite, und in dieser Vorlesung bei der E-Statik. Auch hier kann man Fourier entwickeln.

Als Beispiel wollen wir eine mit a diskretiserte Funktion $f(x)$ betrachten, zunächst definiert für $0 \leq x \leq L/2$ mit $f(0) = 0 = f(L/2)$. Dabei muss natürlich $L/2$ zum Gitter gehören, $N = L/a$ also gerade sein. Obwohl uns nur dieser Bereich interessiert, weiten wir die Definition aus auf $[-L/2, L/2]$, indem wir setzen $f(-x) = -f(x)$ und dann auf die gesamte reelle Achse durch periodische Fortsetzung. Nun ist f antisymmetrisch periodisch mit Periode L. Damit gilt die Entwicklung (2.36), wobei allerdings nur die Sinus Anteile auftreten

$$f(x) = \sum_{n=1,\ldots,N/2-1} \beta_n \sin\left(\frac{\pi}{L/2} nx\right). \tag{2.37}$$

Wir sehen hier, dass durch $N/2 - 1$ Werte β_n gerade die ebensovielen unabhängigen Funktionswerte $f(a), f(2a), \ldots, f(L/2 - a)$ dargestellt werden. Das einzige, was sich für $a \to 0$ ändert, ist, dass die Summe über $n = 1, 2, \ldots, \infty$ läuft (Saite, Dirichlet Nullrandbedingung).

Wenn stattdessen die Ableitung verschwindet, diskret etwa $f(0) = f(a)$ und $f(L/2) = f(L/2+a)$ so bietet sich die Fortsetzung zu einer symmetrisch periodischen Funktion an und man bekommt die Cosinus Reihe (Neumann Randbedingung).

2.11 Matlab

Will man die "langsame" FT implementieren, so ist nur zu beachten, dass Formeln mit den mathematisch vorteilhaften Indizes $n = 0, 1, \ldots, N-1$ auf Matlab Vektoren abgebildet werden müssen, die mit $1, 2, \ldots, N$ zu indizieren sind, obwohl trivial, eine beliebte Fehlerquelle.

In Matlab werden selbstverständlich 'state of the art' FFT Routinen zur Verfügung gestellt. Sie heissen `fft` und `ifft`, die Inverse Transformation. Die Konventionen passen nicht perfekt zu den hier (nicht ohne Grund) getroffenen. Die Übersetzung ist aber nicht schwer. Für die hier behandelten Vektoren $F(n), \tilde{F}(m)$ gilt

$$F = N \times \texttt{ifft}(\tilde{F}), \quad \tilde{F} = (1/N) \times \texttt{fft}(F).$$

Die Matlab Routinen funktionieren für alle N, vermutlich sind Zweierpotenzen aber besonders schnell.

3 Eindimensionale Quantenmechanik mit Matrixmethoden

In diesem Teil wollen wir einige eindimensionale quantenmechanische Systeme untersuchen. Wir werden die Operatoren der betrachteten Quantensysteme durch Diskretisieren auf Matrizen abbilden und dann die lineare Algebra Funktionalität von MATLAB verwenden. Es handelt sich dabei um einen naiven direkten Ansatz, der amüsante Veranschaulichungen von Lehrbuchproblemen ermöglicht. Eine Verallgemeinerung auf mehr als zwei oder drei Dimensionen ist aber schwierig, da Matrizen dann schnell extrem groß werden. Solche Systeme bis hin zur Quantenfeldtheorie werden in einer Formulierung über das Feynman'sche Pfadintegrale behandelt. Für die hier betrachteten 'kleinen' Systeme führt unsere Methode aber zu numerisch wesentlich effektiveren Rechnungen.

3.1 Diskretisierte Operatoren

In der Schrödinger Darstellung der Quantenmechanik sind Zustände durch Wellenfunktionen $\psi(x)$ gegeben, wobei normalerweise $x \in (-\infty, \infty)$. Operatoren können durch Integralkerne dargestellt werden. Also eine Abbildung $\phi = \hat{A}\psi$ z. B. durch

$$\phi(x) = \int_{-\infty}^{\infty} \mathrm{d}y A(x,y)\psi(y). \tag{3.1}$$

Für den einfachen Ortsoperator $\hat{x}$, der $\psi(x)$ einfach mit x multipliziert, hat man etwa

$$A(x,y) = x\delta(x-y). \tag{3.2}$$

A kann als Matrix angesehen werden, deren Indizes x, y kontinuierlich sind und einen unendlichen Bereich haben. Daraus muß zur numerischen Behandlung wieder eine endliche Matrix werden, die aber bis auf unvermeidliche kleine Abweichungen das System physikalisch repräsentieren muss.

Klarerweise muß dann der x-Bereich endlich gemacht und dann noch diskretisiert werden. Da viele Systeme Symmetrie unter Spiegelungen haben, ist es günstig, diese in unserer Behandlung so weit wie möglich zu respektieren. Wir schränken ein auf $x \in (-L/2, L/2)$, symmetrisch zum Zentrum. Für welche L ist dies physikalisch sinnvoll? Wir denken überwiegend an Systeme vom Typ harmonischer Oszillator. Dort haben alle Eigenzustände eine Breite, jenseits derer sie exponentiell abfallen (klassisch verbotener Bereich).

Die Breite ist durch Parameter im Hamilton Operator bestimmt. Wenn L deutlich größer als diese ist, können wir erwarten, daß dort künstlich eingeführten Ränder oder periodische Randbedingungen nicht viel ausmachen. Wählt man angepaßte Einheiten, in denen die erwähnte physikalische Skala von der Ordnung Eins ist, so bedeutet dies $L \gg 1$, z. B. $L = 10$, und der Einfluß ist durch Variieren von L zu untersuchen. Ganz ähnliche Überlegungen gelten für das Diskretisieren mit einer Schrittweite a, die fein genug sein muß, um Strukturen in den vorkommenden $\psi(x)$ auflösen zu können. Bei natürlichen Einheiten[10] heißt das $a \ll 1$.

Wir führen nun ein

$$x_k = ka, \quad k = -M, \dots, -1, 0, 1, \dots, M, \quad N = 2M + 1, \tag{3.3}$$

das sind N Punkte auf der x-Achse. Wellenfunktionen ersetzen wir durch Vektoren der Länge N mit (i. a. komplexen) Komponenten $\psi_k = \psi(x_k)$, $k = -M, \dots, M$. Wenn wir identifizieren (s. auch unten) $L = Na$, dann gilt $x_{\min} = x_{-M} = -(L - a)/2$ und $x_{\max} = x_M = (L - a)/2$. Obwohl der Rand im Sinne der Approximation keine Rolle spielen soll, werden wir eine Vorschrift benötigen (z. B. für diskretisierte Differentialoperatoren), wo man hinkommt, wenn man z. B. von x_{-M} in negativer Richtung geht. Wir wählen hier periodische Randbedingungen mit Periodenlänge L, d. h. wegen $(-M - 1)a + L = Ma$ ist x_M der linke Nachbarpunkt von x_{-M}. Man kann auch sagen, Indizes k und $k \pm N$ sind identifiziert (Modulobildung). Wir haben sozusagen das Intervall zu einem Ring zusammengebogen.

Der Ortsoperator ist natürlicherweise nun durch die Diagonalmatrix

$$\hat{x}_{kl} = x_k \delta_{kl}, \quad -M \le k, l \le M \tag{3.4}$$

gegeben. Eine einfache Idee, die unseren früheren Näherungen für Ableitungen entspricht, würde die kinetische Energie (minus 2. Ableitung in der Schrödinger Darstellung) darstellen mit Hilfe von

$$(\hat{p}^2\psi)_k = \frac{\hbar^2}{a^2}(2\psi_k - \psi_{k+1} - \psi_{k-1}). \tag{3.5}$$

Man käme bald auf die Idee bessere Näherungen für $\psi''(x_k)$ durch Verwendung von 5 Punkten usw. zu verwenden. Da wir von einer dünnen Besiedelung

[10] Wir nehmen an, daß es nur *eine* physikalische Länge im Problem gibt wie $\sqrt{\hbar/m\omega_0}$ beim Oszillator. In ihren Vielfachen wird dann gemessen.

der Matrix von $\hat{p}^2$ im folgenden keine großen Vorteile haben werden, wollen wir hier einen etwas anderen Weg beschreiten. Wir nutzen dabei die mit der Fourier Analyse gelieferte Interpolation aus.

Für "Funktionen" ψ_k über diskretisiertem Definitionsbereich gibt es die diskrete Fourierentwicklung, wie im Detail besprochen wurde. Man muss die Formeln nur an unsere wegen Symmetrien gewählte Indizierung hier anpassen und, wie schon geschehen, ungerade N nehmen. Die entscheidende Identität (im Fourier Kapitel Masterformel genannt und bewiesen) ist

$$\delta_{kl} = \frac{1}{N}\sum_p \exp[ip(x_k - x_l)], \tag{3.6}$$

wobei die p-Summe geht über N aufeinanderfolgende Werte mit Abstand $2\pi/L$ geht. Auch diese wählen wir symmetrisch,

$$p = \frac{2\pi}{L}j; \quad j = -M, \ldots, M; \quad p \in \left(-\frac{\pi}{a}, \frac{\pi}{a}\right). \tag{3.7}$$

Das so dargestellte Kronecker δ ist N-periodisch in k, l. Schreiben wir $\psi_k = \sum_l \delta_{kl}\psi_l$ mit der Fourier Darstellung, so können wir momentan die linke Seite als in x_k kontinuierlich interpoliert auffassen und in dieser Größe normal differenzieren. Damit kommt man zu folgender $N \times N$ Matrixdarstellung ($\hbar = 1$),

$$(\hat{p}^2)_{kl} = \frac{1}{N}\sum_p p^2 \exp[ip(x_k - x_l)], \tag{3.8}$$

und dies ist die Form, die wir für den (quadrierten) Impulsoperator verwenden wollen. Wenn wir die Antisymmetrie des sin-Anteils beachten, können wir auch reell schreiben

$$(\hat{p}^2)_{kl} = \frac{1}{N}\sum_p p^2 \cos(p(x_k - x_l)). \tag{3.9}$$

3.2 Oszillator Niveaus numerisch

Nachdem wir Matrixdarstellungen vom diagonalen $\hat{x}$ und von $\hat{p}^2$ haben können wir Systeme von Typ

$$\hat{H} = \frac{1}{2}\hat{p}^2 + V(\hat{x}) \tag{3.10}$$

untersuchen (m=1 durch Wahl der Einheiten). Wir beginnen wieder mit dem harmonischen Oszillator,

$$V(\hat{x}) = \frac{1}{2}\hat{x}^2 \tag{3.11}$$

in geeigneten Einheiten. In diesem Fall ist alles exakt bekannt (für $a = 0$ und $L = \infty$). Hier interessieren uns die Energieeigenwerte

$$E_n = n + 1/2, \quad n = 0, 1, \ldots \tag{3.12}$$

Zum Grundzustand gehört die Eigenfunktion

$$\psi^{(0)}(x) \propto \exp(-x^2/2), \tag{3.13}$$

die, wie schon diskutiert, in unseren Einheiten eine Breite von 1 hat. Bei den höheren Zuständen wird dieser Gauß Faktor noch mit Polynomen multipliziert.

Das nun folgende MATLAB Programm setzt die Geometrie und berechnet die Hamilton Matrix H:

```
%
%  file ho_fou.m
%
%
%  Erzeugung Diskretisierung und
%  Hamilton Operator fuer QM Harmonischen Oszillator
%  mit Fourier Ableitung
%  Aufruf [a,x,H] = ho_fou(L,N)
%
function [a,x,H] = ho_fou(L,N)

if ~mod(N,2), error('ho_fou.m: N muss ungerade sein');end
M=(N-1)/2;
%  Raum:
a = L/N;                % Diskretisierungslaenge
x=      a*(-M:M);       % x-Werte
p=(2*pi/L)*(-M:M);      % Impulse

%  kinetischer Teil des Hamilton:
A = zeros(N,N);
```

```
for i=1:N, A(i,:) = x(i) - x; end              % Matrix (x_i - x_j)
H = zeros(N,N);
for i=1:N, H = H + p(i)^2*cos(p(i)*A); end     % kein i*sin wg. p<->-p Symm.
H = H*(0.5/N);                                  % (-1/2) Laplace

%  Potential, Oszillator Teil des Hamilton:
H=H+0.5*diag(x.^2);
%
disp(' H.O. Hamilton mit Fourier-diskretisiertem Laplace Operator: ')
fprintf(' Raumintervall [%g , %g] mit %i Punkten \n',[-L/2 L/2 N]);
```

Obwohl man an Matrizen der Größe O(100) denkt, wollen wir uns zum Zweck des Anschauens die Ausgabe für $L = 6, N = 7$ und damit $a = 6/7$ mal ansehen:

```
[a,x,H] =ho_fou(6,7)
 H.O. Hamilton mit Fourier-diskretisiertem Laplace Operator:
 Raumintervall [-3 , 3] mit 7 Punkten

a =

    0.8571


x =

   -2.5714   -1.7143   -0.8571         0    0.8571    1.7143    2.5714


H =

    5.4994   -1.3121    0.2796   -0.0642   -0.0642    0.2796   -1.3121
   -1.3121    3.6626   -1.3121    0.2796   -0.0642   -0.0642    0.2796
    0.2796   -1.3121    2.5606   -1.3121    0.2796   -0.0642   -0.0642
   -0.0642    0.2796   -1.3121    2.1932   -1.3121    0.2796   -0.0642
   -0.0642   -0.0642    0.2796   -1.3121    2.5606   -1.3121    0.2796
    0.2796   -0.0642   -0.0642    0.2796   -1.3121    3.6626   -1.3121
   -1.3121    0.2796   -0.0642   -0.0642    0.2796   -1.3121    5.4994
```

```
diary off
```

Wie es sich gehört ist H hermitesch, hier reell symmetrisch. Wir verwenden nun die MATLAB Routine **eig** um Eigenwerte und -vektoren dieser 7×7 Matrix anzusehen:

```
[v,d]=eig(H)

v =

    0.0290   -0.0777   -0.2817    0.3048    0.5550    0.3342   -0.6333
    0.1607   -0.3859   -0.5367    0.5140   -0.0214   -0.4309    0.2947
    0.4814   -0.5874   -0.1448   -0.3780   -0.3252    0.3762   -0.1098
    0.6950   -0.0000    0.4724    0.0000    0.4141   -0.3497   -0.0000
    0.4814    0.5874   -0.1448    0.3780   -0.3252    0.3762    0.1098
    0.1607    0.3859   -0.5367   -0.5140   -0.0214   -0.4309   -0.2947
    0.0290    0.0777   -0.2817   -0.3048    0.5550    0.3342    0.6333


d =

    0.4995         0         0         0         0         0         0
         0    1.5067         0         0         0         0         0
         0         0    2.4389         0         0         0         0
         0         0         0    3.7007         0         0         0
         0         0         0         0    4.0529         0         0
         0         0         0         0         0    5.8278         0
         0         0         0         0         0         0    7.6119

diary off
```

Die Matrix `d` enthält die Eigenwerte. Wir erkennen die unteren drei Werte $1/2, 3/2, 5/2$ mit erstaunlicher Genauigkeit. Die jeweils darüberstehenden Spalten von `v` sind die Eigenvektoren. Wir sehen die korrekten (Anti)symmetrieeigenschaften unter Reflektion. Die weiteren Eigenwerte sind weniger genau, und es kommen sogar zwei symmetrische Eigenfunktionen hintereinander. Dies sind das unendliche Problem verzerrende Effekte von endlichem L und/oder a.

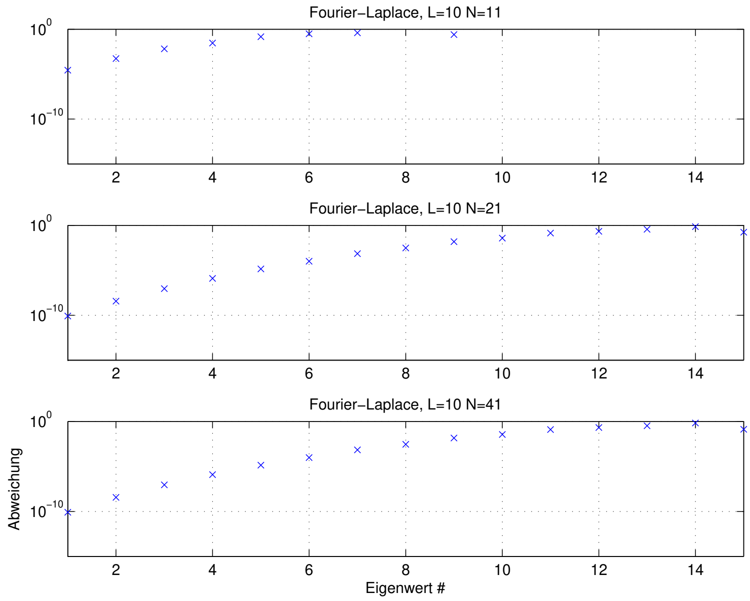

Abbildung 1: Abweichung im Spektrum des harmonischen Oszillators.

Nun wollen wir größere Realisierungen wählen. Für verschiedene Wahlen von L und N plotten wir die Abweichung der numerischen Eigenwerte von den exakten in Abb.1 und Abb.2. Dazu wurden die Eigenwerte aus **eig** mit **sort** geordnet. Wir interpretieren die Bilder wie folgt: Für gegebene Systemgröße bekommt man eine gewisse Anzahl von unteren Anregungen genau. Die höheren spüren irgendwann den "Käfig" und werden verzerrt, egal wie fein man diskretisiert. Bei $L = 10$ bringen mehr als 21 Punkte ($a \approx 1/2$) nichts mehr. $L = 20$ und $N = 101$ ist offenbar keine schlechte Wahl, mit der man ca. 30 Eigenwerte auf 10^{-10} genau hat, die unteren 25 sogar maschinengenau. Dann nehmen die Abweichungen rasch zu, die Zustände werden zu breit. Die Diagonalisierung bei $N = 101$ braucht etwa 0.01 Sekunden auf einem Linux-PC mit Intel Pentium 4 CPU/2.66 GHz.

Mit dem folgenden Stück code kann man auch die Eigenfunktionen bekommen, ordnen und dann plotten:

```
%
%  file evek.m
%
```

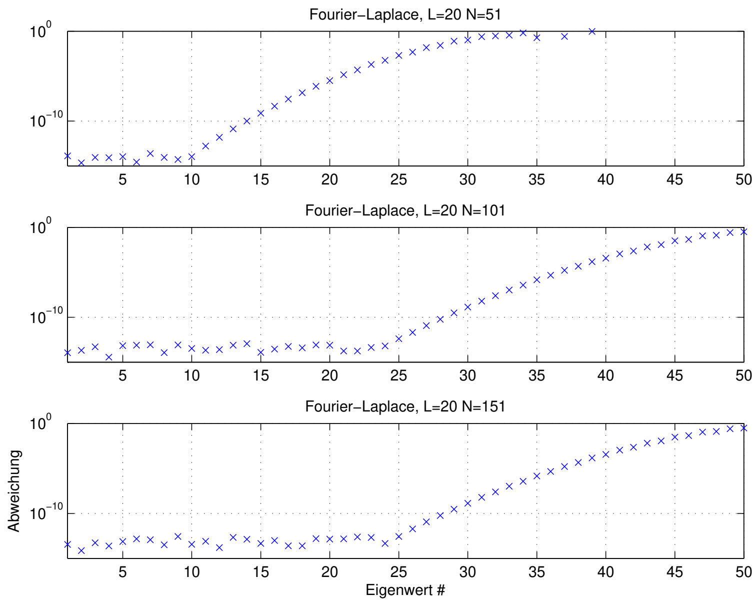

Abbildung 2: Abweichung im Spektrum des harmonischen Oszillators.

```
%  Untersuchung Eigenvektoren des Hamilton
%
%clear all;
clf;

L=20; N=101;
[a,x,H] = ho_fou(L,N);

[v,d]=eig(H);          % Eigenwerte und -vektoren (spalten von v)
sp=diag(d);            % Spalte von Eigenwerten

% ordnen:
[sp,t]=sort(sp);
v=v(:,t(:));

subplot(3,1,1), plot(x,v(:,1:3)); title('n=0,1,2')
subplot(3,1,2), plot(x,v(:,26) ); title('n=25')
subplot(3,1,3), plot(x,v(:,51) ); title('n=50')
```

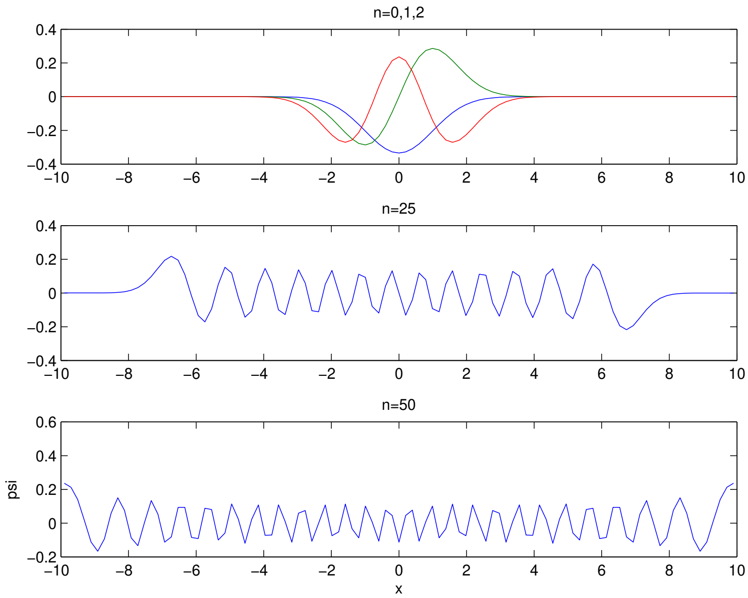

Abbildung 3: Einige numerische Wellenfunktionen ($L = 20, N = 101$)

```
xlabel('x'); ylabel('psi')
```

Das Resultat ist in Abb.3 zu sehen. Es wird deutlich, daß bei $n = 50$ starke Effekte des endlichen Volumens sind und bei $n = 25$ gerade noch nicht. Erstaunlich ist, daß der zugehörige Eigenwert nur einen relativen Fehler von 10^{-13} aufweist, obwohl Diskretisierungseffekte der Wellenfunktion schon deutlich sichtbar sind.

Wieso bekommen wir die Eigenwerte so genau? Für die Volumeneffekte hatten wir schon argumentiert: Solange die Wellenfunktion $|\psi_n(x)|$ im unendlichen System noch exponentiell klein ist bei $x = \pm L/2$, sollte der dadurch verursachte Fehler auch nur exponentiell klein sein. Nun spielen aber Ort und Impuls beim harmonischen Oszillator eine völlig symmetrische Rolle. Dies gilt auch in unserer Diskretisierung, d.h. in der Impulsraumbasis würde p ein Multiplikationsoperator wie (3.4) vorher, und x^2 sähe analog zu (3.8) aus. Der Grundzustand in unseren Einheiten ist auch im Impulsraum Gaussisch mit Breite Eins und $2\pi/a$ wäre dort statt L das endliche 'Volumen' und als solches irrelevant bis auf exponentiell kleine Effekte, solange die Zustände nicht (im Impulsraum) durch Anregung zu sehr verbreitert sind. Dies hängt auch mit dem Abtasttheorem (vgl. Kapitel Fourier) zusammen. Angepasst an die Sprache hier, wo es nicht um Zeitreihen geht: Diskretisierungsfehler entstehen nur, soweit nicht das volle Fourierspektrum überdeckt ist, ansonsten wird das Kontinuumsverhalten exakt repräsentiert. Es gibt hier zwar kein exaktes Analogon zur Nyquist Frequenz bzw. ω_{max}, aber in den unteren Zuständen sind Wellenzahlen größer als π/a eben fast völlig abwesend. Diese Situation wird qualitativ gleich bleiben für andere Bindungsprobleme als den harmonischen Oszillator. Mit dem 'einfachen' Impulsoperator (3.5) gäbe es hingegen Fehlerterme der Ordnung a^2.

3.3 Zeitabhängige Probleme

Nachdem wir nun einige Kontrolle über den Oszillator Hamilton Operator dargestellt als endliche Matrix haben, wollen wir ihn benutzen, um Wellenfunktionen in der Zeit evolvieren zu lassen. Schließlich hat H ja die Doppelbedeutung als Energie Observable und als Generator der Dynamik in der Schrödinger Gleichung

$$i\hbar\frac{\partial}{\partial t}\psi(t) = H\psi(t). \tag{3.14}$$

Wir arbeiten also im Schrödinger Bild (zeitabhängige Zustände, zeitunabhängige Operatoren). Unter $\psi(t)$ kann man sich entweder einen abstrakten Hilbert Raum Vektor oder eine Wellenfunktion $\psi(t,x)$ oder (am besten hier) gleich unsere diskretisierte genäherte Form $\psi_k(t)$ als N komponentigen nun zeitabhängigen Vektor vorstellen. Formal integriert ist (3.14) äquivalent zu (nun wieder $\hbar = 1$)

$$\psi(t) = \exp(-itH)\psi(0), \tag{3.15}$$

wobei man sich die Matrix Exponentiation in der Eigenbasis, wo H diagonal ist, denkt. Man kann die Matrix `v` von Eigenvektoren des letzten Abschnitts dazu verwenden zwischen Orts- und Eigenbasis hin- und herzutransformieren oder aber alles von der MATLAB Routine **expm** erledigen lassen. Diese exponenziert Matrizen im gerade diskutierten Sinn.

Wir betrachten nun das folgende Programm:

```
%
% file zeit.m
%
% Zeitentwicklung einer Gauss Funktion beim Harmonischen Oszillator
%
```

```
clear all; hold off; clf;
L=20; N=101;
[a,x,H] = ho_fou(L,N);                % Hamilton

I = sqrt(-1);
T0 = 2*pi;                             % Periode
T  = T0/10;                            % Strosboskop Zeit

U = expm(-I*T*H);                      % Evolution ueber T

x_z=3; sigma=1/sqrt(3);
psi = exp(-(x-x_z).^2/(2*sigma^2)).'; %Gauss Anfangszustand
psi = psi/norm(psi);                   %Normieren
gauss=psi;
p = psi.*conj(psi);                    % Wahrscheinlichkeit

plot(x,p,'-'); hold;
xlabel('x'); ylabel('p');
pause
%gtext('0');
% gtext erlaubt die Kurven interaktiv per Maus zu bezeichnen;
% es stopp nach jeder Kurve und verlangt input! s. help
t=0;
for i=1:5
  psi = U*psi; t=t+T;
  p = psi.*conj(psi);
  plot(x,p,'-');
  %gtext(num2str(i));
  pause(1);
end
disp('PAUSE'); pause;
hold off;clf;
% Zeitentwicklung des x-Mittelwertes
for i=1:100
  xmean(i) = real((psi'.*x)*psi); % psi-kreuz x psi; sowieso reell
  tplot(i)=t;
  psi=U*psi; t=t+T;
end
```

```
plot(tplot/T0,xmean); hold; plot(tplot/T0,xmean,'x')
xlabel('t/T0'); ylabel('<x>');
```

Als Anfangs Zustand wird hier ein Gauß Paket $\exp[-(x-x_z)^2/(2\sigma^2)]$ verwendet dessen Maximum bei $x_z = 3$ und dessen Breite durch $\sigma^2 = 1/3$ gegeben ist. Dies ist also ein Zustand, der enger als der Grundzustand ist. Er ist sicher kein Energie Eigenzustand. Er wird mit der Zeit komplex, und wir plotten daher

$$p_k(t) = p(t, x_k) \propto |\psi_k(t)|^2, \quad \sum_k p_k = 1. \tag{3.16}$$

Die Oszillator Periode ist $T_0 = 2\pi$ in den gewählten Einkeiten. Die n'te Eigenkomponente oszilliert also $\propto \exp[-i(n+1/2)t]$, so daß $p_k(t)$ diese Periodizität hat unabhängig vom Anfangszustand. Wir bilden $U(T) = \exp(-iTH)$ mit $T = T_0/10$ und plotten die p_k für $t = 0, T, 2T \ldots T_0/2$, danach läuft man durch die gleichen Zustände wieder zurück. Man erhält hier ein stroboskopisches Bild des Quantensystems. Im Programm `zeit.m` wird nur eine Wellenfunktion ψ gespeichert. Man läßt sie dann jeweils mit $\psi \to U(T)\psi = \exp(-iTH)\psi$ um einen Schritt evolvieren (dabei haben wir H zuvor für $L = 20$ und $N = 101$ berechnet). Das Ergebnis wird dann gleich in ein Bild geplottet (siehe Abb.4). Es läßt sich beim Oszillator zeigen, daß eine Gauß'sche Wellenfunktion unter Zeitentwicklung stets Gaußisch bleibt[11]. Allerdings verändert sich, wie man sieht, i. a. die Breite (außer $\sigma = 1$). Nach $T_0/2$ ist das Teilchen klassisch am Umkehrpunkt, und hier ist die anfängliche Form der Wahrscheinlichkeitsverteilung wieder erreicht (nicht aber die Phasen in ψ), und die gleichen Verteilungen werden nun rückwärts angenommen bis die volle Periode durchlaufen ist.

Die Zeitentwicklung des x Mittelwertes über einige Perioden entspricht der klassischen Bewegung. Die Berechnung des Mittelwertes wird im zweiten Teil des Programms **zeit.m** durchgeführt. Die Zeitwentwicklung des Mittelwertes ist in der Abb.5 als Funktion von T/T_0 dargestellt.

---

[11] Der exakte Integralkern zu $U(T)$ ist Gaußisch, und die Konvolution von Gauß Integralen führt wieder auf solche.

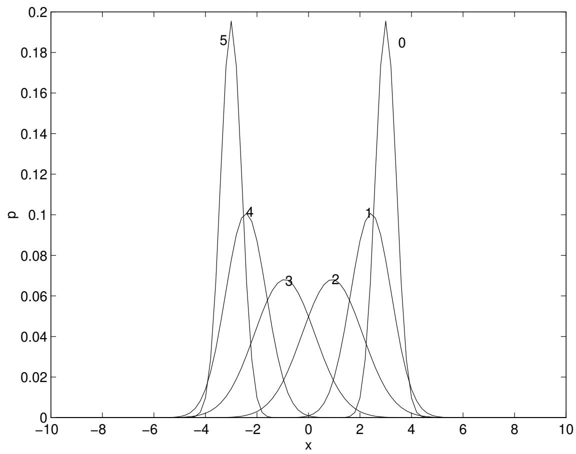

Abbildung 4: Zeitentwicklung von $|\psi|^2$ beim Gauß Paket ($L = 20$, $N = 101$).

3.4 Anharmonischer Oszillator

Es ist nun ein Leichtes, den harmonischen Term durch anharmonische zu ersetzen. Hier nehmen wir zum Beispiel mal willkürlich

$$\hat{H} = \frac{1}{2}(\hat{p}^2 + \hat{x}^4). \tag{3.17}$$

Dieser Hamilton Operator ist leicht auf viele Art und Weise numerisch zu behandeln, nicht aber analytisch. Wieder mit $L = 20, N = 101$ bekommen wir für die untersten 8 Eigenwerte

```
format long
ev(1:8)

ans =

   0.530181045242001
   1.899836514900598
   3.727848968993103
   5.822372755688942
   8.130913009424923
  10.619186459117861
  13.264235591841359
```

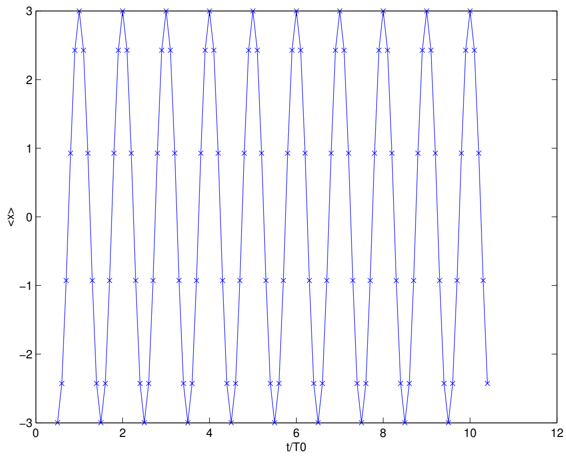

Abbildung 5: Zeitentwicklung des x-Mittelwertes ($L = 20$, $N = 101$).

```
  16.049298855484199

diary off
```

In dem man auch mal $L = 30$ nimmt, kann man abschätzen, daß der zu erwartende relative Fehler von 10^{-12} (Grundzustand) steigt auf etwa 10^{-7} beim größten gezeigten Wert. Durch Steigern von N kommt man zum Schluß, daß der Diskretisierungsfehler hier vernachlässigbar ist. Das Spektrum für diesen Fall ist deutlich verschieden vom harmonischen Oszillator. Die Niveaus sind nicht mehr äquidistant und alle höher.

3.5 Tunneln: stationäre Zustände

Ein wichtiges quantenmechanisches Problem, bei dem man weitgehend auf Näherungsverfahren oder Numerik angewiesen ist, stellt der Tunneleffekt dar. Als gängiges Beispiel mit einem Freiheitsgrad sei hier das System

$$\hat{H} = \frac{1}{2}\hat{p}^2 + V(\hat{x}) \tag{3.18}$$

mit dem Doppelmulden–Potential

$$V(x) = (x - x_{min})^2 (x + x_{min})^2 / (8x_{min}^2) \tag{3.19}$$

behandelt. $V(x)$ hat zwei Minima $V(x) = 0$ bei $x = \pm x_{min}$, sie sind durch eine Barriere der Höhe

$$V(0) = \frac{1}{8} x_{min}^2 \tag{3.20}$$

getrennt. Der Normierungsfaktor von V ist so gewählt, daß $V''(\pm x_{min}) = 1$, d.h. jede Mulde für sich betrachtet ähnelt einem harmonischen Oszillator der Frequenz $\omega = 1$ (in natürlichen Einheiten).

Die Programme `dm_fou.m` und `dmevek.m` erledigen die Konstruktion des Hamilton–Operators und seine Diagonalisierung, es sind offensichtliche Abwandlungen der Routinen, die für den (an-)harmonischen Oszillator geschrieben wurden. Für drei verschiedene Werte von x_{min} erhält man so die 10 niedrigsten Eigenwerte:

```
D.M. Hamilton mit Fourier-diskretisiertem Laplace Operator:
Raumintervall [-10 , 10] mit 151 Punkten

     xmin = 2           xmin = 3           xmin = 5

     0.350239           0.460383           0.489498
     0.523186           0.473929           0.489498
     1.113986           1.132844           1.421839
     1.729921           1.369160           1.421932
     2.469131           1.863507           2.265520
     3.290646           2.386312           2.270646
     4.184444           2.981769           2.934615
     5.141611           3.629893           3.030664
     6.155675           4.325681           3.455764
     7.221563           5.064278           3.790578
```

Im ersten Fall sind die beiden Minima kaum getrennt ($V(0) = .5$), dagegen liegt im dritten Beispiel eine klare Tunnelsituation vor, denn es gibt Eigenwerte weit unterhalb der Schwelle von $V(0) = 3.125$, sie bilden enge Dubletts nahe bei den Eigenwerten des entsprechenden harmonischen Oszillators. Das mittlere Beispiel zeigt ebenfalls diese Struktur, aber weniger extrem, und wird im nächsten Abschnitt zur Illustration der Zeitentwicklung benutzt.

Einige Eigenfunktionen sind in Abb.6 und 7 dargestellt. Im Tunnelfall ($x_{min} = 5$) ist zu sehen, daß die niedrigen Eigenfunktionen abwechselnd durch gerade und ungerade Überlagerungen von Oszillator–Moden in den beiden Minima angenähert werden, bei $x_{min} = 2$ ist das natürlich nicht der Fall.

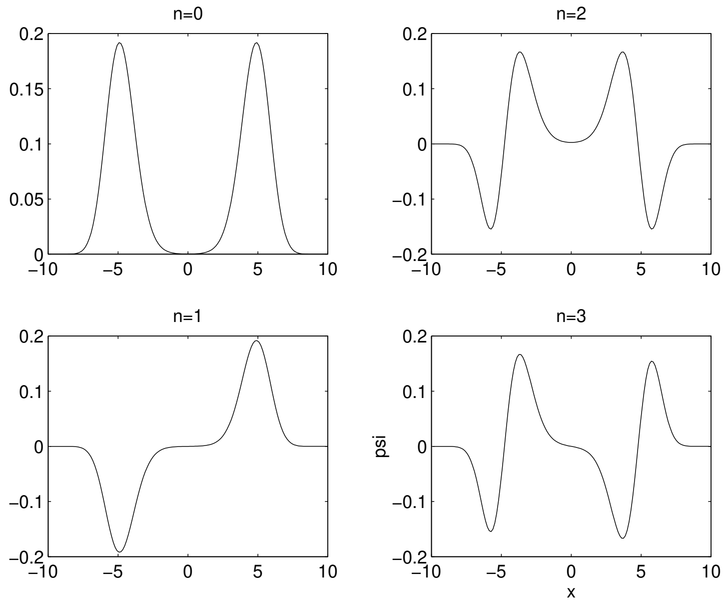

Abbildung 6: Die vier niedrigsten Eigenfunktionen für $x_{min} = 5$

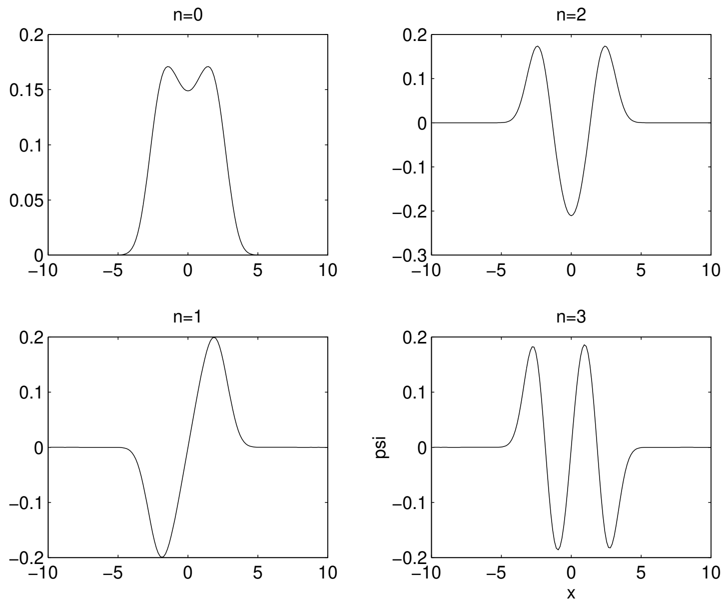

Abbildung 7: Die vier niedrigsten Eigenfunktionen für $x_{min} = 2$

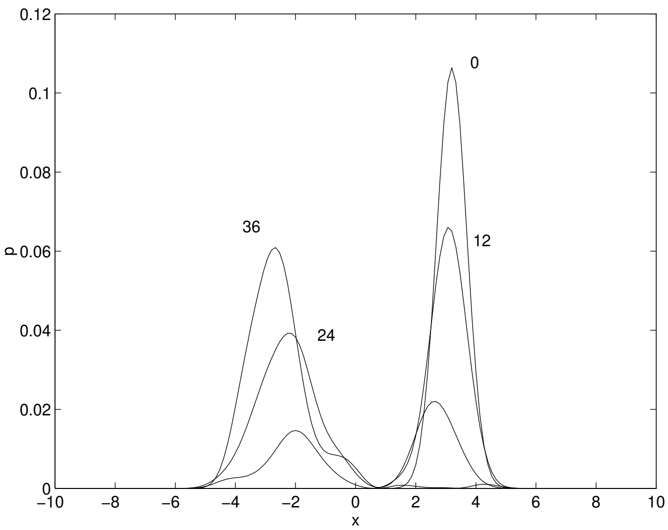

Abbildung 8: $p(x,t) = |\psi(x,t)|^2$ zu vier verschiedenen Zeiten t/T_0

3.6 Tunneln: Zeitentwicklung

Wir wollen nun, wie schon zuvor beim harmonischen Oszillator, den diskretisierten Hamiltonoperator benutzen, um den Zeitentwicklungsoperator

$$U(t) = \exp(-itH) \tag{3.21}$$

zu bilden und damit Lösungen der zeitabhängigen Schrödingergleichung darzustellen. Dies ist im Programm `dmzeit.m` implementiert. Als Anfangszustand dient wieder ein (normiertes) Gauß–Paket $\sim \exp[-(x-x_z)^2/(2\sigma^2)]$. In Abb.8 ist die Entwicklung der Aufenthaltswahrscheinlichkeit dargestellt für ein Tunnelpotential mit $x_{min} = 3$. Für den Anfangszustand ist $\sigma = 1/\sqrt{2}$ und $x_z = 3.2$ angesetzt, d.h. man erwartet schon ein oszillierendes Verhalten in der einfachen Mulde – der Grundzustand hätte $\sigma = 1$ und $x_z = 3$. Dem ist nun ein Übergang in die andere Mulde überlagert. Die zeitliche Entwicklung des Mittelwerts $\angle x\rangle$ in Abb.9 zeigt das noch deutlicher.

Die Tunnelzeit kann man quantitativ durch folgende einfache Überlegung verstehen: Wenn der Anfangszustand näherungsweise als Überlagerung der beiden niedrigsten Eigenfunktionen dargestellt wird, d.h. $\psi(x, t = 0) \sim \psi_0(x) + \psi_1(x)$ nach Abb.6, dann folgt eine Zeitentwicklung

$$\psi(x,t) \;\sim\; e^{-iE_0t}\psi_0(x) + e^{-iE_1t}\psi_1(x)$$

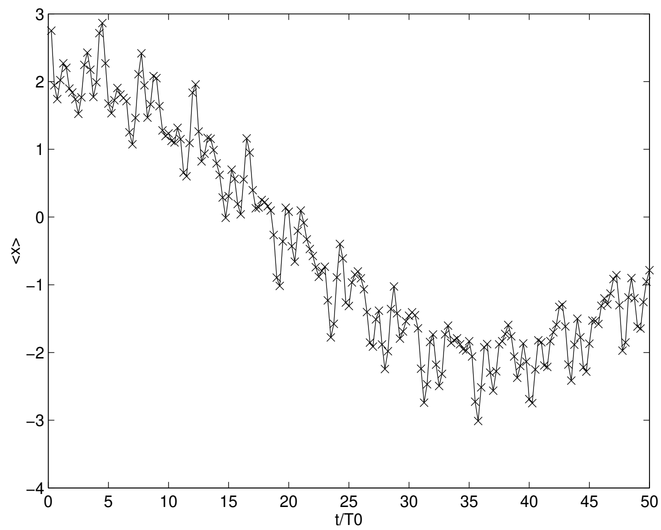

Abbildung 9: Zeitentwicklung von $< x >$ beim Tunneln

$$= \quad e^{-iE_0 t}\left(\psi_0(x) + e^{-i(E_1-E_0)t}\psi_1(x)\right). \tag{3.22}$$

Ein Übergang in die andere Mulde liegt vor, wenn $\psi(x,\tau) \sim \psi_0(x) - \psi_1(x)$, also zuerst bei

$$\tau = \frac{\pi}{E_1 - E_0}. \tag{3.23}$$

Mit der im vorigen Abschnitt angegebenen Aufspaltung ergibt sich $\tau = 73.8\pi$ oder $\tau/T_0 = 36.9$ ($T_0 = 2\pi$ war die Oszillatorperiode), in guter Übereinstimmung mit Abb.9.

Wenn bei der Zerlegung des Anfangszustands höhere Eigenfunktionen wesentlich beitragen, dann werden diese Komponenten schneller tunneln, weil $E_3 - E_2 > E_1 - E_0$ usw. Die Wellenfunktion wird sich bald auf beide Mulden verteilen, das Tunnelbild ist dann weniger deutlich.

4 Quantenmechanische Streuung in einer Dimension

In diesem Abschnitt wollen wir Streulösungen der Schrödinger Gleichung in einer Dimension diskutieren. Wie üblich wiederholen wir die Begriffsbildungen aus P3 bzw. Quantenmechanik I, reproduzieren numerisch Resultate für die dort typischerweise behandelten einfachen Potentialstufen, um dann zu Allgemeinerem fortzuschreiten.

4.1 Wellenpakete mit Matrix Quantenmechanik

Wir wollen zunächst mit Hilfe der Matrix Quantenmechanik aus dem vorherigen Kapitel die Ausbreitung eines freien Teilchen nachbilden. Wir wählen Einheiten mit $\hbar = 1$, müssen also nicht zwischen Impulsen und Wellenzahlen unterscheiden, und setzen auch $m = 1$. Wir präparieren eine Gauss Anfangswellenfunktion

$$\psi(x, t = 0) \propto \int dk \exp(-(b^2/2)(k - k_0)^2 + ikx) \propto \exp\Big(-\frac{x^2}{2b^2} + ik_0 x\Big), \quad (4.1)$$

zentriert um Impuls k_0 mit Breite b im Ort. Im endlichen diskretisierten System gibt es zwar nur diskrete Impulse $p \in (-\pi/a, \pi/a)$ im Abstand $2\pi/L$, die Gauss Integrale werden aber gut genähert solange $a \ll b \ll L$ gilt. Es gibt nun zwei Zeitskalen, wenn diese Anfangswellenfunktion mit dem freien Hamiltonoperator evolviert

$$\psi(t) = \exp\Big(-it\,\frac{1}{2}\,\hat{p}^2\Big)\psi(0). \quad (4.2)$$

Das ist zum Einen die Zeit $T = L/k_0$ die auch klassisch zur Bewegung durch das gegebene Intervall der Länge L benötigt wird. Zum anderen wissen wir, dass das Paket quantenmechanisch breitfliesst. Die Skala, auf der das passiert ist durch $t_f \sim b^2$ gegeben. Um den Weg des Wellenpakets durch unser Volumen zu verfolgen sollte zumindest nicht $T \gg t_f$ gelten, sonst verlöre das Paket zu schnell seine 'Identität'. Wir betrachten folgenden Kompromiss,

$$b = \sqrt{aL}, \quad k_0 = \frac{1}{2}\pi/a, \quad (4.3)$$

der $a/b = b/L = \sqrt{a/L} = 1/\sqrt{N}$ und $T \sim t_f$ ergibt. Die Evolution in Schritten von $T/10$ ist in Fig.10 im oberen Bild zu sehen mit $L = 20$ und

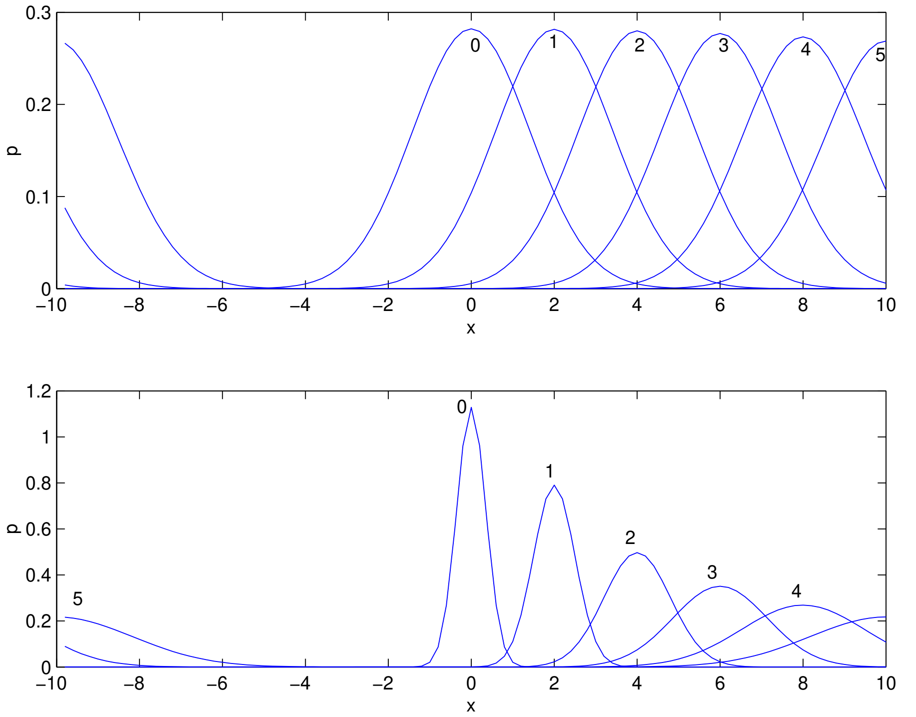

Abbildung 10: Evolution eines freien Wellenpakets mit zwei verschiedenen Breiten.

$N = 101$. Gezeigt ist jeweils die Wahrscheinlichkeitsdichte $p(x,t) \propto |\psi(x,t)|^2$ in Kontinuum Normierung, $\sum_x a\, p(x) = 1$. Im unteren Bild ist $b \to b/4$ ersetzt. Deutlich werden auch die periodischen Randbedingungen[12].

Als Nächstes bauen wir eine Potentialstufe der Höhe V_0 und der Breite $2w = 2$ symmetrisch über $x \in (-w, +w)$ auf. Mit $L = 40$ und $N = 201$ wird ein Wellenpaket der Breite $\sqrt{aL}/2$ startend bei $x = -13$ von links mit Impuls $k_0 = \pi/(4a)$ einlaufen gelassen. Fig.11 zeigt die Situation mit $V_0 = k_0^2$ (kinetische Energie $V_0/2$), Fig.12 bei $V_0 = 0.7 \times k_0^2/2$.

[12]Sie sind durch die Fourier Ableitung impliziert. Im letzten Abschnitt waren Wellenfunktionen bei $\pm L/2$ praktisch Null und der Rand spielte keine Rolle.

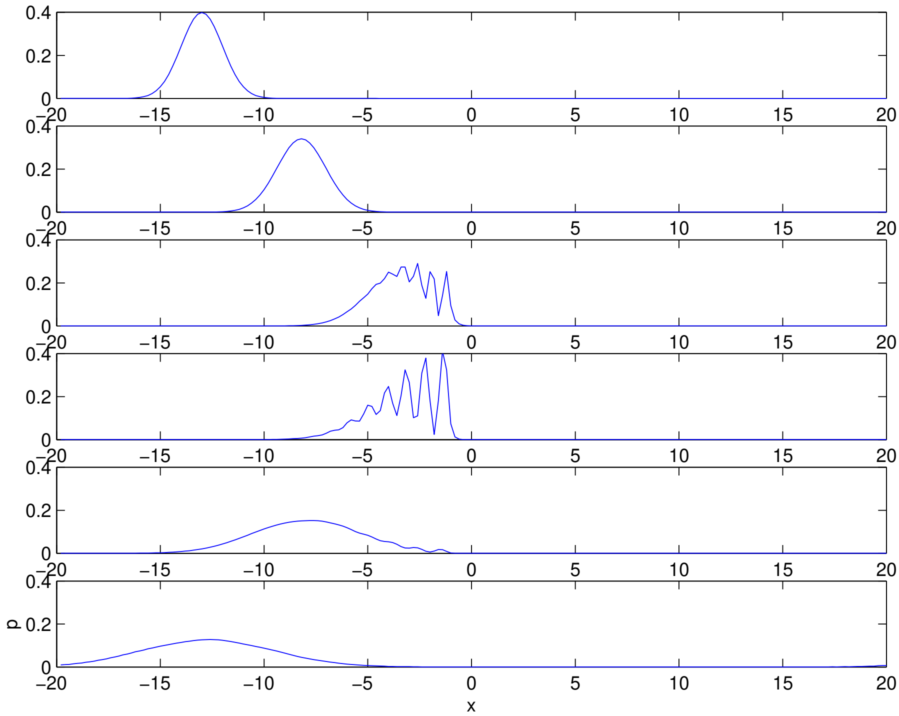

Abbildung 11: Streuung an einer Potentialstufe $V_0 = 2E$.

4.2 Stationäre Streulösungen

In diesem Abschnitt diskutieren wir "normale" QM auf der reellen Linie, $a = 0, L = \infty$.

Bei den eben betrachteten Lösungen der zeitabhängigen Schrödinger Gleichung (frei und mit Potential) bestand die Anfangs Konfiguration aus überlagerten ebenen Wellen $\exp(ikx)$ mit k nahe einem $k_0 > 0$ (nur dort gab es nennenswerte Amplituden), die ein nach rechts laufendes Wellenpaket bildeten. Es handelt sich hier nicht um Energieeigenzustände. Solche sind im freien Fall allerdings die einzelnen Komponenten,

$$H \exp(ikx) = \frac{1}{2}\hat{p}^2 \exp(ikx) = E \exp(ikx), \quad E = \frac{k^2}{2}. \tag{4.4}$$

Es handelt sich jedoch um (für $L = \infty$) uneigentliche Elemente des Hilbertraums, da sie nicht normierbar sind.

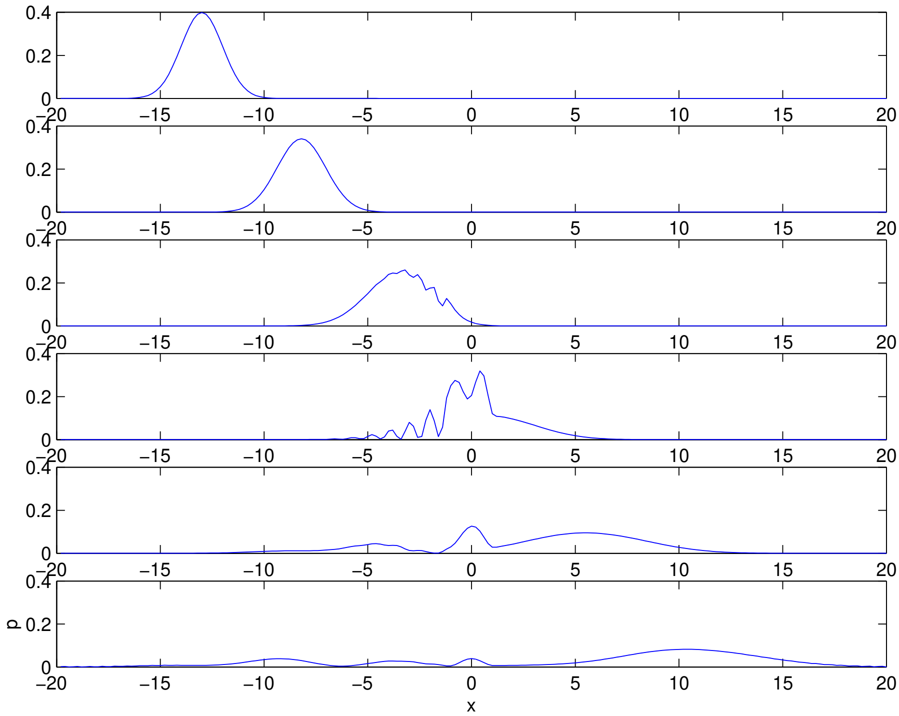

Abbildung 12: Streuung an einer Potentialstufe $V_0 = 0.7E$.

Es ist nun nützlich, auch mit Potential auf solchen Zuständen aufzubauen. Wir nehmen an, dass gilt $V(x) = 0$ für $|x| \geq w$ und Paritätssymmetrie $V(x) = V(-x)$ mit ansonsten beliebigem $V(x)$. Streulösungen mit definierter Parität sind gerade und ungerade Lösungen von

$$\left\{-\frac{1}{2}\frac{d^2}{dx^2} + V(x)\right\}\psi_k^{\pm}(x) = E(k)\psi_k^{\pm}(x), \quad \psi_k^{\pm}(-x) = \pm\psi_k^{\pm}(x) \tag{4.5}$$

auf dem *gesamten* Raum. Für $x \leq -w$ und $x \geq w$ müssen die $\psi_k^{\pm}(x)$ aus ebenen Wellen bestehen und sie sind auf beiden Seiten trivial durch die Spiegel (Anti)symmetrie miteinander verbunden. Wir können Norm und Phase fixieren in dem wir verlangen dass für $x \leq -w$ die Form

$$\psi_k^{\pm}(x) = \exp(ikx) \pm C_{\pm}(k)\exp(-ikx), \quad E(k) = k^2/2 \tag{4.6}$$

gilt, wobei beide Lösungen für jedes k existieren. Auf $x \in [-w, w]$ ist die

Lösung i. a. nicht explizit bekannt. Die Vorzeichen sind so, dass $V \equiv 0$ zu $C_\pm = 1$ führt.

Wenn man aus $\psi_k^\pm$ Wellenpakete bildet, so haben wir nicht ganz die physikalische Situation, sondern zu frühen Zeiten laufen von links *und* rechts Pakete spiegelbildlich zueinander ein (Amplituden Eins) und zu späten Zeiten in Gegenrichtung wieder aus (Amplituden C_+ bzw. C_-). Wegen Strom- bzw. Wahrscheinlichkeitserhaltung gilt für die Intensitäten

$$|C_+(k)|^2 = 1 = |C_-(k)|^2, \tag{4.7}$$

$C_\pm$ sind also Phasen,

$$C_\pm = \exp(2i\delta_\pm), \quad \delta_\pm \in (\pi/2, \pi/2]. \tag{4.8}$$

Die physikalische Streulösung ϕ_k kann nun aus ihnen kombiniert werden als

$$\phi_k(x) = \frac{1}{2}(\psi_k^+(x) + \psi_k^-(x)), \tag{4.9}$$

so dass die physikalischen Randbedingungen

$$\phi_k(x) = \begin{cases} \exp(ikx) + r(k)\exp(-ikx) & \text{für } x < -w \\ t(k)\exp(ikx) & \text{für } x > +w \end{cases} \tag{4.10}$$

erfüllt sind mit

$$t(k) = \frac{1}{2}(C_+(k) + C_-(k)), \quad r(k) = \frac{1}{2}(C_+(k) - C_-(k)). \tag{4.11}$$

Die Phasen $C_\pm$ enthalten also sämtliche Information wie das Potential sich in Streuexperimenten mit einlaufendem Strahl und entferntem Detektor auswirkt.

4.3 Streuung und endliche Volumen Effekte

Man könnte nun die Streulösungen $\psi_k^\pm$ bzw. ϕ_k numerisch konstruieren und daraus Streuphasen ableiten. Es wurde jedoch gezeigt, dass man auch aus dem Einfluss eines endlichen (periodischen) Volumens – wie es zur numerischen Behandlung ja unvermeidlich ist – indirekt und elegant Information über Streuung bekommen kann. Einen solchen Zusammenhang gibt es allgemein in der Quantenfeldtheorie, und er ist von Lüscher im Detail ausgearbeitet worden [6, 7]. Bei Simulationen zur Elementarteilchen Physik ist dies äußerst interessant, da dieser Trick die einzige Möglichkeit ist, solche Daten aus Simulationen zu extrahieren. Man zieht also Nutzen aus einer vermeintlich lästigen Beschränkung.

Betrachten wir zunächst den Fall ohne Potential. Dann führt die Bedingung von Periodizität zur Quantisierung von Wellenzahlen und damit der Energie

$$\exp(ikL/2) = \exp(-ikL/2) \Rightarrow k = \frac{2\pi}{L}n, \quad n = 0, \pm 1, \pm 2, \dots \tag{4.12}$$

Mit Potential ist die Bedingung $\psi_k^+(-L/2) = \psi_k^+(+L/2)$ trivial wegen der Symmetrie, aus der Periodizität der Ableitung folgt jedoch die Bedingung

$$C_+(k) \exp(ikL) = 1. \tag{4.13}$$

Beim antiperiodischen Fall ist der Wert selbst nichttrivial und erfordert

$$C_-(k) \exp(ikL) = 1. \tag{4.14}$$

Auch dies sind Quantisierungsbedingungen für k, die nun aber die Streuphasen beinhalten,

$$\exp[2i\delta_\pm(k) + ikL] = 1. \tag{4.15}$$

Nach Lüscher können wir nun folgende numerische Prozedur durchführen:

- Man berechne im L-periodischen Volumen die Eigenwerte E_j des Hamilton Operators. In unserem Fall konstruieren wir direkt die Matrix $\hat{H}$ durch genügend feine Diskretisierung ($a \ll L$ und $ak \ll 1$ für betrachtete k).
- Setze $k_j = \sqrt{2E_j}$.
- Für diese Werte k_j kennen wir nun $\exp[2i\delta_\pm(k_j)] = \exp[-ik_jL]$.
- Je nach Symmetrie der zugehörigen Wellenfunktion haben wir δ_+ oder δ_- gefunden.
- Falls $\delta_\pm$ noch für andere k gesucht sind, so kann man L variieren.

Als Beispiel können wir nun die Potentialstufe

$$V(x) = V_0\theta(w - |x|) \tag{4.16}$$

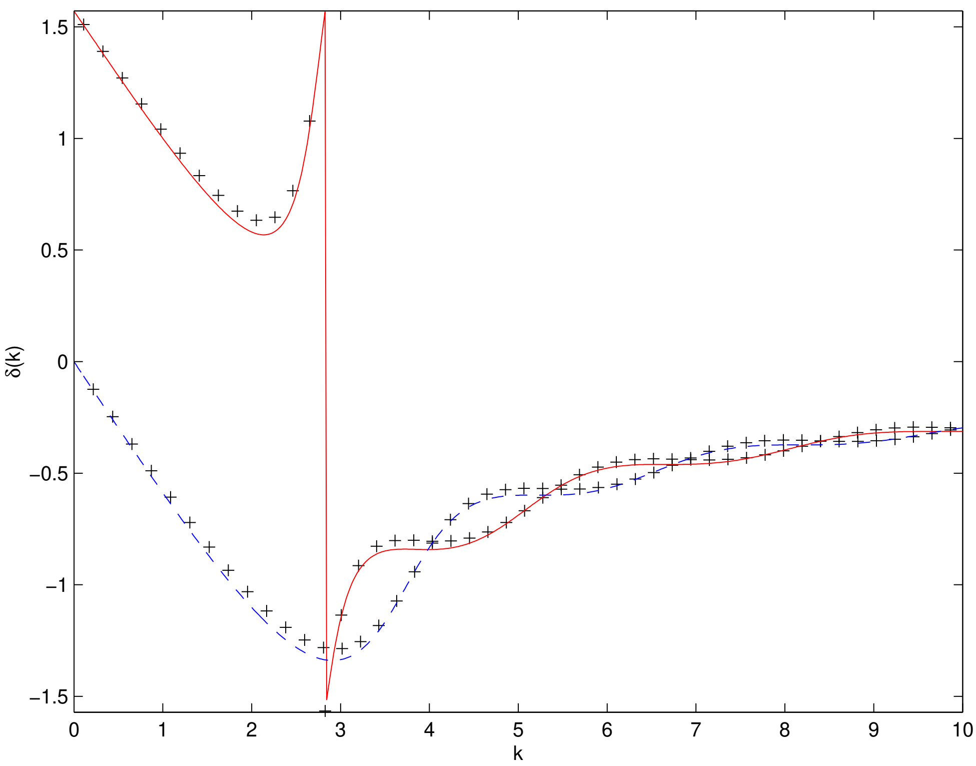

Abbildung 13: Streuphasen $\delta_\pm$ (links oben $+$, unten $-$) für das Kastenpotential, exakt und aus Energien im endlichen Volumen.

betrachten (θ=Stufenfunktion). In diesem Fall lautet die symmetrische (antisymmetrische) Lösung im Innenbereich $\cos(k'x)$ $(\sin(k'x))$ mit

$$k' = \sqrt{k^2 - 2V_0} \tag{4.17}$$

mit positiv imaginärem k' bei $k^2 < 2V_0$. Dieses Beispiel wird in jedem QM Buch behandelt, und aus der Stetigkeit der Wellenfunktion und ihrer ersten Ableitung (ψ, ψ') bei $\pm w$ erhält man

$$C_+ = \exp(-2ikw)\frac{k + ik'\tan(k'w)}{k - ik'\tan(k'w)} \tag{4.18}$$

und

$$C_- = -\exp(-2ikw)\frac{k - ik'\cot(k'w)}{k + ik'\cot(k'w)}. \tag{4.19}$$

In Fig.13 sehen wir als Beispiel das Kastenpotential der Höhe $V_0 = 3$, Breite $2w = 2$ im periodischen Volumen $L = 30$. Die Kurven für δ_+ (ausgezogen) und δ_- (gestrichelt) stammen von den Formeln (4.18) und (4.19), während die Datenpunkte von den numerischen Eigenwerten von H mit Diskretisierung $N = 201$ kommen. Besonders stark variieren die Phasen nahe $k \approx 2.5$, wo $k^2/2 \approx V_0$ gilt. Die Qualität der Näherung variiert nicht sehr uniform mit N, was vermutlich daran liegt, dass unsere Diskretisierung der Ableitung nichtlokal ist oder dass die Potentialsprünge hohen Fourier Komponenten entsprechen und schwierig aufzulösen sind.

Die Phasen $\delta_\pm$ haben noch eine etwas andere Interpretation. Stellen wir uns ein Zweiteilchenproblem vor mit Potential $V(|x_1 - x_2|)$ zwischen Teilchen identischer Masse $m_1 = m_2 = 2$. Wenn man nun wie üblich die freie Schwerpunktbewegung absepariert, so können wir unser Teilchen mit der effektiven Masse $1/m = 1/m_1 + 1/m_2 = 1$ als Bewegung der Relativkoordinate $x = x_1 - x_2$ interpretieren. Unterliegen die beiden Primärteilchen der Bose Statistik, so sind nur die ('Paritäts') geraden Zustände zu nehmen und δ_+ ist die Zweiteilchen Streuphase. Entsprechendes gilt für den fermionischen Fall. In diesem Bild ist der Zusammenhang zwischen endlichen Volumen Effekten und der Streuphase besonders plausibel. Da die Teilchen gemeinsam auf einem Ring eingesperrt sind, müssen sie sich mit einer Frequenz $\propto 1/L$ begegnen und aneinander streuen. Durch diese Wechselwirkung wird die Energie gegenüber dem freien Fall verschoben, was durch die Streuphasen in der so extrem einfachen Beziehung (4.15) quantitativ beschrieben wird.

Offensichtlich können wir nun recht effektiv die Streuphasen für beliebige Potentiale numerisch über das endliche Volumen studieren, solange eine hinreichend gute Diskretisierung möglich ist. Ein Beispiel einer allgemeineren solchen Studie in einer 2-dimensionalen Quantenfeldtheorie ist in [8] zu finden, wo eine theoretisch unter gewissen Annahmen vorgeschlagene Form der S-Matrix numerisch überprüft wurde.

4.4 Born'sche Näherung.

Selbst in einer Dimension ist es eine Ausnahme, wenn man die Streuphasen geschlossen berechnen kann wie beim Stufenpotential. Wie wollen daher hier die verbreitete Born'sche Näherung aufstellen. Die können wir dann mit diversen numerisch gewonnenen Streuphasen vergleichen (z. B. in Übungen). Meist wird die Näherung über die Lipmann-Schwinger Integralgleichung konstruiert, die der Schrödinger Gleichung zusammen mit den Randbedingungen äquivalent ist. Für unsere Zwecke (nur führende Ordnung) folgt hier eine einfachere Ableitung.

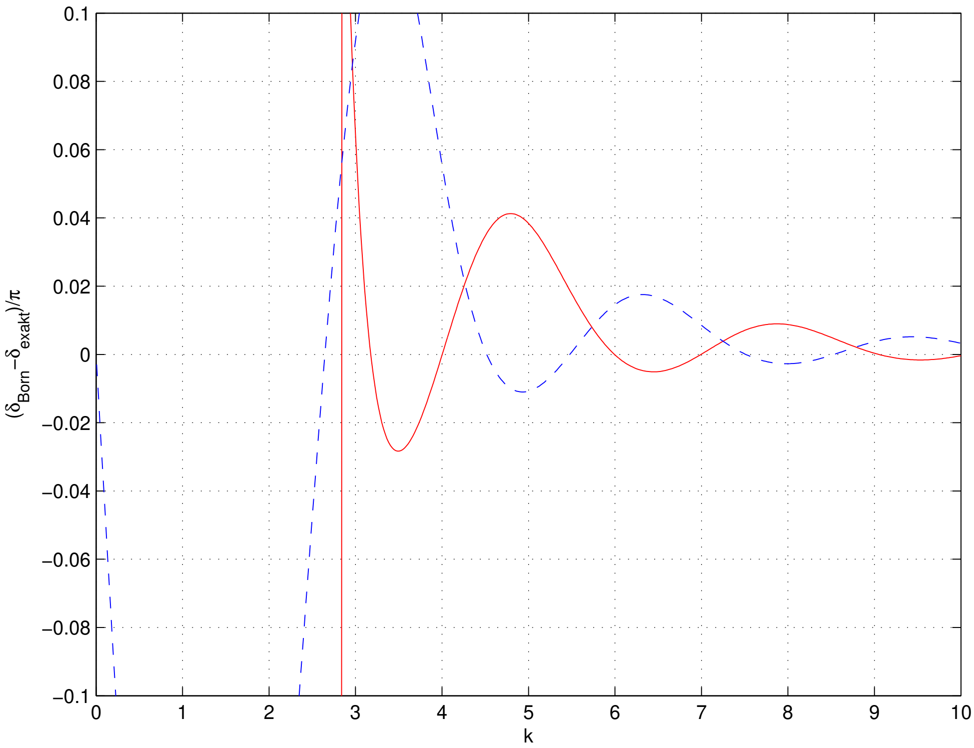

Abbildung 14: Differenz der exakten Streuphasen zu denen der Born'schen Näherung (durch π) für die Potentialstufe (Parameter wie zuvor).

Wir setzen für die Streulösungen an

$$\psi_k^{\pm} = \psi_{k,0}^{\pm} + \psi_{k,1}^{\pm} + \ldots \tag{4.20}$$

Dabei sind

$$\psi_{k,0}^{+} = 2\cos(kx), \quad \psi_{k,0}^{-} = 2i\sin(kx) \tag{4.21}$$

die Lösungen mit korrekten Randbedingungen und ohne Potential ($C_{\pm} = 1$). Dann soll $\psi_{k,1}^{\pm}$ das Potential in erster Ordnung berücksichtigen. Es muss dazu die Gleichung

$$\frac{1}{2}\left\{\frac{d^2}{dx^2} + k^2\right\}\psi_{k,1}^{\pm} = V\psi_{k,0}^{\pm} \tag{4.22}$$

lösen. Wegen unserer Randbedingungen ist $\psi^{\pm}_{k,1}$ nur eine Modifikation der *gestreuten* Welle. Es gilt daher

$$\psi^{\pm}_{k,1}(x) = \pm 2i\delta_{\pm}\exp(-ikx) \quad \text{für} \quad x \leq -w, \tag{4.23}$$

wobei $\delta_{\pm}(k)$ von erster Ordnung in V ist. Um sie zu berechnen bilden wir

$$\int_{-X}^{+X} dx\, \psi^{\pm}_{k,0} \left\{ \frac{d^2}{dx^2} + k^2 \right\} \psi^{\pm}_{k,1} = 2\int_{-w}^{+w} dx\, V(x)(\psi^{\pm}_{k,0})^2 \tag{4.24}$$

mit dem Integrationsbereich $X \geq w$. Wenn wir links zweimal partiell integrieren, bekommen wir als Resultat

$$\left[\psi^{\pm}_{k,0}\frac{d}{dx}\psi^{\pm}_{k,1} - \psi^{\pm}_{k,1}\frac{d}{dx}\psi^{\pm}_{k,0}\right]_{-w}^{+w} = \mp 8k\delta_{\pm}.$$

Damit ergibt sich in Born'scher Näherung

$$\delta_{+} = -\frac{2}{k}\int_0^{+w} dx\, V(x)\cos^2(kx) \tag{4.25}$$

$$\delta_{-} = -\frac{2}{k}\int_0^{+w} dx\, V(x)\sin^2(kx). \tag{4.26}$$

Das ergibt für die Stufe

$$\delta_{\pm} = -\frac{V_0 w}{k}\left(1 \pm \frac{\sin(2kw)}{2kw}\right). \tag{4.27}$$

Die Abweichungen gegen die exakten Phasen sehen wir in Fig.14. Es handelt sich offenbar um eine gute Näherung für hohe Energie, aber nicht für den Bereich bei $k \approx 2.5$ wo “viel los” ist und resonanzartige Phänomene auftreten. Bei großen k hingegen stellt die Wirkung des Potential ein kleine Störung der durchlaufenden Wellen dar und die Streuphasen durch unser abstoßendes Potential haben kleine negative Werte.

5 Diffusion

In diesem Abschnitt geht es um die eindimensionale Diffusion. Hier geht es um das Lösen von Anfangswertaufgaben für Dichten im Gegensatz zu Randwertaufgaben für Felder in der Elektrostatik. Die Diffusionsgleichung ähnelt formal der Schrödingergleichung ohne i und $\hbar$ (also reell), und das Analogon der Wellenfunktion (nicht ihr Quadrat) besitzt die physikalische Interpretation einer Dichte. Im Gegensatz zur stets (in ψ) linearen Schrödingergleichung werden wir uns aber auch für nichtlineare Diffusion interessieren. Die räumliche Struktur stellt neue Anforderungen an die algorithmische Stabilität der diskreten Zeitentwicklung bei Euler-artigen Lösungsverfahren.

5.1 Diffusionsgleichung

Wir betrachten ein zeitabhängiges Feld in einer Raumdimension, das physikalisch z.B. eine Dichte oder Temperaturverteilung bedeuten mag, und bezeichnen es mit $\phi(x,t)$. Um die Diffusionsgleichung elementar herzuleiten, führen wir eine Diskretisierung in Raum ($x = nh$) und Zeit ($t = n\tau$) ein – für eine numerische Behandlung wird das früher oder später ohnehin nötig sein. Die Änderung des Feldes $\phi(x,t) \to \phi(x,t+\tau)$ bei x ist nun gegeben durch " Übergänge" $x \leftrightarrow x+h$ mit "Wahrscheinlichkeit" $w(x+h/2)$ und $x \leftrightarrow x-h$ mit $w(x-h/2)$ und proportional zur Differenz der "Konzentration" zwischen x und seinen nächsten Nachbar Punkten. Dazu kommt eine lokale "Quelle" $q(x,t)$:

$$\begin{aligned}\phi(x,t+\tau) = \phi(x,t) \quad &+ \quad w(x+h/2)\,[\phi(x+h,t)-\phi(x,t)] \\ &+ \quad w(x-h/2)\,[\phi(x-h,t)-\phi(x,t)] \\ &+ \quad q(x,t). \end{aligned} \tag{5.1}$$

Hier erkennt man die genäherten Ableitungen

$$\phi(x,t+\tau)-\phi(x,t) \;=\; \tau\frac{\partial}{\partial t}\phi(x,t) + \mathrm{O}(\tau^2) \tag{5.2}$$

$$\phi(x\pm h,t)-\phi(x,t) \;=\; \pm h\frac{\partial}{\partial x}\phi(x\pm h/2,t) + \mathrm{O}(h^3) \tag{5.3}$$

und erhält

$$\tau\frac{\partial}{\partial t}\phi(x,t) \;\approx\; w(x+h/2)\,h\frac{\partial}{\partial x}\phi(x+h/2,t)$$

$$
\begin{aligned}
&- \quad w(x-h/2)\, h\frac{\partial}{\partial x}\phi(x-h/2,t) + \mathrm{O}(h^3) \\
&+ \quad q(x,t) \\
&\approx \quad h^2 \frac{\partial}{\partial x}\left[w(x)\frac{\partial}{\partial x}\phi(x,t)\right] + q(x,t).
\end{aligned} \tag{5.4}
$$

So entsteht die Diffusionsgleichung

$$
\frac{\partial}{\partial t}\phi(x,t) = \frac{\partial}{\partial x}\left[D(x)\frac{\partial}{\partial x}\phi(x,t)\right] + S(x,t) \tag{5.5}
$$

mit (ortsabhängiger) Diffusionskonstante

$$
D(x) = \frac{h^2}{\tau} w(x) \tag{5.6}
$$

und Quellterm

$$
S(x,t) = \frac{1}{\tau} q(x,t). \tag{5.7}
$$

Für den Fall $D = \text{const}$, $S = 0$ und $x \in \mathbb{R}$ läßt sich eine wichtige Lösung von Glg.(5.5) für alle $t > 0$ angeben:

$$
G(x,t) = \frac{1}{\sigma(t)\sqrt{2\pi}} \exp\left[-\frac{x^2}{2\sigma(t)^2}\right]. \tag{5.8}
$$

Dies ist eine (normierte) Gauß–Verteilung mit zeitabhängiger Breite

$$
\sigma(t) = \sqrt{2Dt}. \tag{5.9}
$$

Wegen

$$
G(x,t) \to \delta(x) \quad \text{für} \quad t \to 0^+ \tag{5.10}
$$

spielt $G(x,t)$ die Rolle einer Green Funktion für die Diffusionsgleichnung: das Anfangswertproblem ($\leftrightarrow \phi(x,0)$ gegeben) wird allgemein gelöst durch

$$
\phi(x,t) = \int_{-\infty}^{\infty} dy\, G(x-y,t)\phi(y,0) \quad \text{für} \quad t > 0. \tag{5.11}
$$

Wenn nur ein endliches x-Intervall mit Randbedingungen zu betrachten ist, hilft eine räumliche Fourier–Entwicklung weiter. Bei ortsabhängigem $D(x)$ muß man auf numerische Methoden zurückgreifen.

5.2 FTCS–Diskretisierung

Eine diskretisierte Form der Diffusionsgleichung kennen wir schon: wir behalten die Konvention bei, die Diffusionskonstanten auf den Zwischengitterpunkten anzusiedeln, und schreiben Glg.(5.1) zusammen mit (5.6) und (5.7) nun als

$$\begin{aligned} \phi(x,t+\tau) &= \phi(x,t) \\ &+ \frac{\tau}{h^2} D(x+h/2)\,[\phi(x+h,t)-\phi(x,t)] \\ &+ \frac{\tau}{h^2} D(x-h/2)\,[\phi(x-h,t)-\phi(x,t)] \\ &+ \tau S(x,t). \end{aligned} \tag{5.12}$$

Dies hat gerade die Form einer Rekursionsgleichung für die Zeitentwicklung (vgl. Euler-Methode in Kap. "Anfangswertprobleme"). Nach der Art der Ableitungsbildungen wird diese Version unter der Bezeichnung FTCS (Forward Time Centered Space) geführt.

Um den Einfluß der Diskretisierung abzuschätzen[10, Kap.7], nehmen wir wieder $D = const$ und $S = 0$ und setzen für die x-Abhängigkeit ebene Wellen an. Für ein Intervall $0 \le x \le L$ mit Randbedingungen $\phi(0,t) = \phi(L,t) = 0$ sei z.B.

$$\phi(x,t) = a(t)\sin kx \quad \text{mit} \quad k = \frac{\pi n}{L},\ n = 1,2,\ldots. \tag{5.13}$$

Ein beliebiges $\phi(x,0)$ kann aus solchen Sinuswellen überlagert werden; wegen der Linearität der Diffusionsgleichung kann ihre Zeitentwicklung einzeln untersucht werden. Die Kontinuums-Glg.(5.5) wird gelöst durch

$$a(t) = a(0)e^{-Dk^2 t}, \tag{5.14}$$

die diskrete Rekursion (5.12) lautet dagegen

$$a(t+\tau) = a(t) + \frac{\tau}{h^2} D(2\cos kh - 2)a(t) \tag{5.15}$$

$$\Rightarrow a(n\tau) = a(0)\left[1 - 4D\frac{\tau}{h^2}\sin^2(kh/2)\right]^n. \tag{5.16}$$

Es sollte also für alle in Frage kommenden k

$$1 - 4D\frac{\tau}{h^2}\sin^2(kh/2) \approx e^{-Dk^2\tau} \tag{5.17}$$

erfüllt sein. Das gilt nur für die langwelligen Anteile mit $kh \ll 1$, wo der Sinus durch sein Argument ersetzt werden kann, und mehr kann man auf einem diskreten Gitter nicht verlangen. Zur näherungsweisen Exponenzierung für diese Moden ist ausserdem ein hinreichend kleiner Zeitschritt τ erforderlich, so daß gilt

$$Dk^2\tau \ll 1. \tag{5.18}$$

Wenn jedoch

$$D\frac{1}{h^2}\,\tau > \frac{1}{2} \tag{5.19}$$

gilt, dann ist die Lösung der diskreten Gleichung für Moden mit $k \sim \pi/h$ nicht nur ungenau, sondern sogar instabil: es gibt Moden, die nach Glg.(5.16) nicht gedämpft werden, sondern (dem Betrag nach) exponentiell anwachsen! Das führt zu numerischen Problemen, selbst wenn diese Moden in der physikalischen Lösung gar nicht angeregt sind. Zumindest im "Rundungsfehlerrauschen" kommen diese Moden vor und "übernehmen" die Evolution in endlicher Zeit.

5.3 Nichtlineare Diffusion und Selbstorganisation

Im Zusammenhang mit Fragen der Selbstorganisation (Strukturentstehung) und der Funktion biologischer Uhren hat die Gruppe um I. Prigogine (bio)chemische Reaktionszyklen untersucht. Dabei ist man auf folgendes Modell gestoßen: zwei Substanzen X und Y reagieren miteinander gemäß

$$\begin{aligned} A &\rightarrow X \\ B+X &\rightarrow Y+D \\ 2X+Y &\rightarrow 3X \\ X &\rightarrow E, \end{aligned} \tag{5.20}$$

wobei A und B aus einem Reservoir konstanter Dichte stammen sollen, D und E sind Abfallprodukte ohne weiteren Einfluß. Wenn man für X und Y auch noch berücksichtigt, daß sich ihre Konzentration durch Diffusion räumlich verteilt, dann erhält man für die Dichten $X(x,t)$ und $Y(x,t)$ das folgende System von Differentialgleichungen[10, S.189ff]:

$$\begin{aligned} \frac{\partial}{\partial t}X(x,t) &= D_X\nabla^2 X + k_aA - k_bBX + k_cX^2Y - k_dX \\ \frac{\partial}{\partial t}Y(x,t) &= D_Y\nabla^2 Y + k_bBX - k_cX^2Y. \end{aligned} \tag{5.21}$$

Die nichtlineare Kopplung, die das System dynamisch so interessant macht, kommt dadurch zustande, daß die lokale Rate einer chemischen Reaktion dem *Produkt* der Dichten der Reaktionspartner proportional ist.

Hier wollen wir nur den Fall einer Raumdimension betrachten, $0 \le x \le L$. An den Rändern sollen diesmal die Neumann–Bedingungen

$$\frac{\partial}{\partial x}\begin{Bmatrix} X \\ Y \end{Bmatrix}(x,t) = 0 \quad \text{für} \quad x = 0, L \tag{5.22}$$

gelten, die gewährleisten, daß kein Material durch die "Wand" fließt.

Durch geeignete Skalierungen kann man die Systemgröße $L = 1$ setzen und $k_i = 1$ erreichen:

$$\begin{aligned} \frac{\partial}{\partial t}X(x,t) &= D_X\nabla^2 X + A - (B+1)X + X^2Y \\ \frac{\partial}{\partial t}Y(x,t) &= D_Y\nabla^2 Y + BX - X^2Y, \end{aligned} \tag{5.23}$$

so daß sich die Vielzahl von Parametern reduziert auf D_X, D_Y, A und B.

Bevor man sich an die numerische Lösung macht, zunächst noch ein paar analytische Vorbetrachtungen. Die Gleichungen (5.23) haben eine räumlich konstante, statische Lösung

$$X_0 = A \qquad Y_0 = B/A. \tag{5.24}$$

Zur Analyse ihrer Stabilität setzt man $X = X_0 + X'$, $Y = Y_0 + Y'$ und linearisiert:

$$\begin{aligned} \frac{\partial}{\partial t}X'(x,t) &= D_X\nabla^2 X' + (B-1)X' + A^2Y' \\ \frac{\partial}{\partial t}Y'(x,t) &= D_Y\nabla^2 Y' - BX' - A^2Y'. \end{aligned} \tag{5.25}$$

Die Fluktuationen werden als Eigenmoden der Form

$$\begin{Bmatrix} X' \\ Y' \end{Bmatrix}(x,t) = \begin{Bmatrix} \xi \\ \eta \end{Bmatrix} e^{\omega t}\cos kx \quad \text{mit} \quad k = \frac{\pi n}{L},\ n = 0, 1, 2, \ldots \tag{5.26}$$

angesetzt, ein Mode mit $\mathrm{Re}\,\omega > 0$ ist instabil. Einsetzen liefert das Gleichungssystem

$$\begin{aligned} \omega\,\xi &= (B - 1 - D_Xk^2)\xi + A^2\eta \\ \omega\,\eta &= -B\xi - (A^2 + D_Yk^2)\eta. \end{aligned} \tag{5.27}$$

Mit den Abkürzungen

$$\begin{aligned} \alpha &= B - 1 - D_X k^2 \\ \beta &= A^2 + D_Y k^2 \end{aligned}$$

ist dieses nichttrivial lösbar, wenn die quadratische Gleichung

$$\omega^2 - (\alpha - \beta)\omega + A^2 B - \alpha\beta = 0 \tag{5.28}$$

erfüllt ist. Ihre Lösungen lauten

$$\omega_\pm = \frac{1}{2}\left\{\alpha - \beta \pm [(\alpha + \beta)^2 - 4A^2 B]^{1/2}\right\}. \tag{5.29}$$

Um das Einsetzen von Instabilitäten systematisch zu diskutieren, nehmen wir D_X, D_Y und A jeweils fest an und variieren B. Bei $B = 0$ setzen wir $\omega_+ = \alpha = -1 - D_X k^2 < 0$ und $\omega_- = -\beta = -A^2 - D_Y k^2 < 0$. Mit zunehmendem B kann nun der Übergang zur Instabilität auf zwei verschiedene Weisen auftreten:

(a) Von zwei rellen Lösungen wird die grössere positiv. Nach Glg.(5.28) ist der Grenzfall dazu erreicht, wenn $\alpha\beta = A^2 B$, d.h. für Wellenzahl k bei

$$B = (1 + D_X k^2)(1 + \frac{A^2}{D_Y k^2}). \tag{5.30}$$

Die rechte Seite hat ein Minimum bei

$$k_c = \left[\frac{A}{\sqrt{D_X D_Y}}\right]^{1/2}, \tag{5.31}$$

mit dieser Wellenzahl (bzw. dem nächstliegenden $k = \pi n/L$) tritt der erste instabile Mode auf, und zwar bei

$$B = B_c = \left[1 + A\sqrt{\frac{D_X}{D_Y}}\right]^2. \tag{5.32}$$

(b) Bei einem Paar konjugiert komplexer Lösungen wird gerade $\mathrm{Re}\omega_\pm = 0$, d.h. $\alpha = \beta$ oder

$$B = 1 + A^2 + (D_X + D_Y)k^2. \tag{5.33}$$

Die Diskriminante $\mathcal{D}$ in Glg.(5.29) kann man schreiben als

$$\mathcal{D} = (\alpha + \beta)^2 - 4A^2 B = (A^2 + B - \Delta_k)^2 - 4A^2 B \tag{5.34}$$

mit

$$\Delta_k = 1 - (D_Y - D_X)k^2 \tag{5.35}$$

$\mathcal{D} < 0$ geht nur mit $\Delta_k > 0$. Weiter kann man umformen

$$\mathcal{D} = [B - (A - \sqrt{\Delta_k})^2][B - (A + \sqrt{\Delta_k})^2]. \tag{5.36}$$

Da der erste Faktor größer ist als der zweite, muss letzterer negativ und ersterer positiv sein. Die Bedingung für komplexe Wurzeln lautet also

$$\Delta_k > 0 \quad \text{und} \quad \left(A - \sqrt{\Delta_k}\right)^2 < B < \left(A + \sqrt{\Delta_k}\right)^2. \tag{5.37}$$

Das ist offenbar immer erfüllt für die kleinste Lösung von (5.33) die sich für $k = 0$ ergibt, $B = B_0 = 1 + A^2$.

Das Auftreten von Instabilität in den linearisierten Gleichungen (5.25) signalisiert, daß das volle (nichtlineare) System (5.23) langfristig nicht mehr in den Gleichgewichtszustand (5.24) übergeht, sondern ein Verhalten ähnlich den instabilen Moden annimmt. Wenn $B_c < B_0$, dann tritt bei $B \simeq B_c$ ein räumlich oszillierendes Muster auf, dessen Wellenzahl durch k_c aus Glg.(5.31) bestimmt ist. Wie ein Vergleich von B_c (5.32) und $B_0 = 1 + A^2$ zeigt, ist das genau dann der Fall, wenn

$$D_Y > D_X \quad \text{und} \quad A > \frac{2}{\sqrt{D_Y/D_X} - \sqrt{D_X/D_Y}}. \tag{5.38}$$

Im anderen Fall $B_0 < B_c$ entsteht ein Langzeit–Verhalten, das sogar räumlich *und* zeitlich oszilliert. Die Komplexität des räumlichen Musters hängt davon ab, für wieviele Wellenzahlen $k = \pi n/L$ die Dichte B die Instabilitätsschwelle überschreitet. Ein schönes Beispiel raum–zeitlicher Oszillation, das durch numerische Integration des Systems (5.23) mit Hilfe des einfachen FTCS–Verfahrens gewonnen wurde, ist in Fig.15 dargestellt. Eine Anleitung zur Implementierung wird in Form einer Übungsaufgabe ausgegeben.

5.4 Implizite Verfahren

Das Instabilitätsproblem der FTCS-Iteration kann man durch einen merkwürdigen Kunstgriff lösen: man nimmt sich vor, die räumlichen Differenzenterme in Glg.(5.12) zur "neuen" Zeit $t + \tau$ anstelle von t zu nehmen. Formal

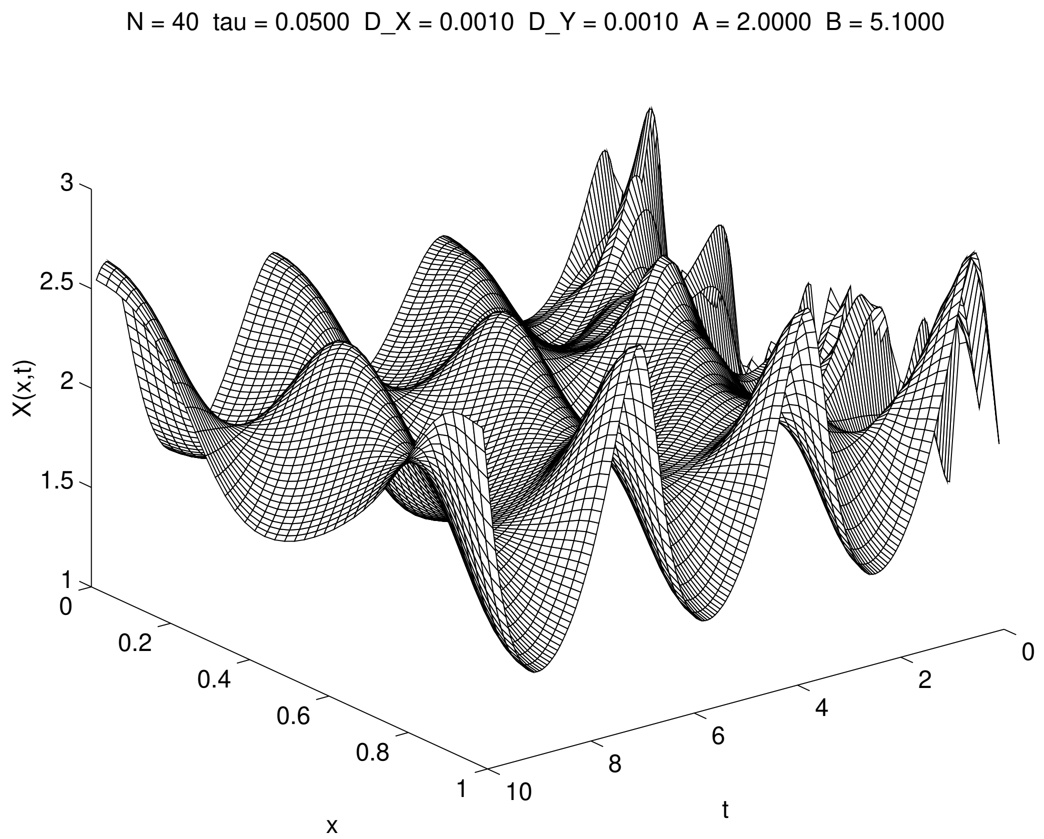

Abbildung 15: Oszillierende Lösung der Reaktionsgleichung

bedeutet das

$$\begin{aligned}
\phi(x,t+\tau) &= \phi(x,t) \\
&+ \frac{\tau}{h^2} D(x+h/2)\left[\phi(x+h,t+\tau)-\phi(x,t+\tau)\right] \\
&+ \frac{\tau}{h^2} D(x-h/2)\left[\phi(x-h,t+\tau)-\phi(x,t+\tau)\right] \\
&+ \tau S(x,t). \qquad (5.39)
\end{aligned}$$

Damit wird bei der Stabilitätsanalyse in Glg.(5.16) der störende Term $\sim \sin^2(kh/2)$ auf die linke Seite gebracht:

$$a(t+\tau) \; + \; \frac{\tau}{h^2} 4D \sin^2(kh/2) a(t+\tau) = a(t) \qquad (5.40)$$

$$\Rightarrow a(n\tau) \;=\; a(0)\left[1+4D\frac{\tau}{h^2}\sin^2(kh/2)\right]^{-n} \tag{5.41}$$

und alle Moden werden nun stabil iteriert. Natürlich hat dieses "Wunder" seinen Preis: die Differenzengleichung (5.39) ist nicht mehr nach $\phi(x,t+\tau)$ aufgelöst, sondern definiert die Iteration nur noch implizit (daher der Name). Man sollte also eher

$$\begin{aligned}
\phi(x,t+\tau) \;&-\; \frac{\tau}{h^2}D(x+h/2)\left[\phi(x+h,t+\tau)-\phi(x,t+\tau)\right] \\
&-\; \frac{\tau}{h^2}D(x-h/2)\left[\phi(x-h,t+\tau)-\phi(x,t+\tau)\right] \\
&=\; \phi(x,t)+\tau S(x,t).
\end{aligned} \tag{5.42}$$

schreiben und dies als lineares Gleichungssystem für $\phi(x,t+\tau)$ bei gegebenem $\phi(x,t)$ auffassen, dabei ist x der Vektorindex.

Eine andere Variante besteht darin, den Laplace–Term "zur Hälfte" auf die linke Seite zu bringen (Crank–Nic[h]olson–Verfahren):

$$\begin{aligned}
\phi(x,t+\tau) \;&-\; \frac{\tau}{2h^2}D(x+h/2)\left[\phi(x+h,t+\tau)-\phi(x,t+\tau)\right] \\
&-\; \frac{\tau}{2h^2}D(x-h/2)\left[\phi(x-h,t+\tau)-\phi(x,t+\tau)\right] \\
&=\; \phi(x,t) \\
&+\; \frac{\tau}{2h^2}D(x+h/2)\left[\phi(x+h,t)-\phi(x,t)\right] \\
&+\; \frac{\tau}{2h^2}D(x-h/2)\left[\phi(x-h,t)-\phi(x,t)\right] \\
&+\; \tau S(x,t).
\end{aligned} \tag{5.43}$$

Hiermit wird die Zeitableitung mit einer Genauigkeit der Ordnung τ^2 approximiert, denn abgesehen vom Quellterm $S(x,t)$ ist die Diskretisierung symmetrisch bezüglich $t+\tau/2$. Die Stabilitätsanalyse liefert diesmal eine Mischform von Glg.(5.16) und (5.41):

$$a(n\tau) = a(0)\left[\frac{1-2D\frac{\tau}{h^2}\sin^2(kh/2)}{1+2D\frac{\tau}{h^2}\sin^2(kh/2)}\right]^n \tag{5.44}$$

und zeigt wieder Stabilität für alle Moden k und beliebige τ.

Es ist anzumerken, daß das FTCS–Verfahren leicht auf Diffusion in mehreren Raumdimensionen auszudehnen ist, bei den impliziten Methoden sind dazu noch weitere Tricks nötig.

5.5 Lösung eines tridiagonalen Gleichungssystems

Die impliziten Rekursionsgleichungen (5.42) und (5.43) sind von der tridiagonalen Form

$$\begin{pmatrix} \beta_1 & \gamma_1 & & & & \\ \alpha_2 & \beta_2 & \gamma_2 & & & \\ & \alpha_3 & \beta_3 & \gamma_3 & & \\ & & \dots & \dots & \dots & \\ & & & \alpha_{N-1} & \beta_{N-1} & \gamma_{N-1} \\ & & & & \alpha_N & \beta_N \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \dots \\ \phi_{N-1} \\ \phi_N \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_{N-1} \\ b_N \end{pmatrix}, \tag{5.45}$$

wobei die Unbekannten $\phi(x, t+\tau)$ hier kurz mit ϕ_i bezeichnet sind und die bekannten Größen auf der rechten Seite mit b_i.

In einem ersten Durchlauf (Gauß–Elimination) werden die unter der Diagonale stehenden α_i beseitigt. Dazu wird die mit α_2/β_1 multiplizierte erste Gleichung von der zweiten subtrahiert, d.h.

$$\begin{aligned} \alpha_2 &\rightarrow 0 \\ \beta_2 &\rightarrow \beta_2' = \beta_2 - \frac{\alpha_2}{\beta_1}\gamma_1 \\ b_2 &\rightarrow b_2' = b_2 - \frac{\alpha_2}{\beta_1}b_1, \end{aligned} \tag{5.46}$$

dann analog (unter Benutzung von β_2' und b_2'!) die zweite Gleichung von der Dritten subtrahiert, so daß $\alpha_3 \rightarrow 0$ usw. Es entsteht

$$\begin{pmatrix} \beta_1' & \gamma_1 & & & \\ & \beta_2' & \gamma_2 & & \\ & & \beta_3' & \gamma_3 & \\ & & & \dots & \dots \\ & & & & \beta_{N-1}' & \gamma_{N-1} \\ & & & & & \beta_N' \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \dots \\ \phi_{N-1} \\ \phi_N \end{pmatrix} = \begin{pmatrix} b_1' \\ b_2' \\ b_3' \\ \dots \\ b_{N-1}' \\ b_N' \end{pmatrix}. \tag{5.47}$$

Dieses System wird nun von unten nach oben gelöst:

$$\begin{aligned} \phi_N &= b_N'/\beta_N' \\ \phi_i &= (b_i' - \gamma_i\phi_{i+1})/\beta_i', \quad i = N-1, N-2, \dots \end{aligned} \tag{5.48}$$

(Rückwärts–Substitution). Diese Prozeduren sind wegen ihres stark rekursiven Charakters tatsächlich leichter zu programmieren als mathematisch aufzuschreiben.

Auf diese Weise verlangen die Iterationen der impliziten Verfahren zwei zusätzliche, einfache Schleifen über die Gitterpunkte, der Aufwand für eine Iteration bleibt $\sim N$, wenn auch mit einem größeren Vorfaktor. Das ist der Preis der Stabilität.

6 Perkolation

Bei der Perkolation handelt es sich darum, daß Systeme, die aus lokal zufälligen Elementen aufgebaut werden, unter bestimmten Umständen zu "Clustern" zusammenklumpen, die einen langreichweitigen Zusammenhang herstellen können. Dieser Mechanismus spielt eine entscheidende Rolle bei so verschiedenen Phänomenen wie der elektrischen Leitfähigkeit von Legierungen, der Ausbreitung von Epidemien und Waldbränden oder der Ergiebigkeit von Erdölfeldern.

In diesem Kapitel wollen wir uns der Perkolation von der Seite der Computersimulation nähern, die umfangreichen theoretischen Konzepte würden eine eigene Vorlesung füllen. Eine detailierte Einführung in Theorie und Numerik geben Stauffer und Aharony[9].

6.1 Typen von Perkolationsproblemen

Das einfachste Perkolationsproblem entsteht, indem man die Punkte eines Gitters unabhängig voneinander mit einer gewissen Wahrscheinlichkeit p "besetzt" und dann nach Clustern fragt, die die besetzten Punkte über ihre Nachbarschaftsbeziehungen bilden (besetzte nächste Nachbarn). Hier spricht man von *Punktperkolation* (site percolation). Die Gitter können verschiedene Dimensionen und Symmetrien (kubisch, dreieckig etc.) haben.

Bei der *Bondperkolation* besitzen die Verbindungslinien (bonds) zwischen den Gitterpunkten die Möglichkeit unabhängig voneinander leitend oder nichtleitend (aktiv/passiv, an/aus,...) zu sein. Cluster bestehen aus über aktive Bonds verbundenen Gitterknoten (sites). Alle Sites werden in Cluster eingeteilt, inklusive Einerclustern.

Schließlich kann man auch auf das Gitter verzichten und z.B. Scheiben, Kugeln o.ä. betrachten, die durch Überlappung zusammenhängen.

In allen Fällen untersucht man, ob bei einem bestimmten Parameterwert von p Perkolation einsetzt, was naiv bedeutet, dass *ein* Cluster existiert, der einen nennenswerten Teil des Gesamtraums bedeckt. Wenn das geschieht, dann findet man Phänomene wie an kritischen Punkten (Phasenübergängen zweiter Ordnung): langreichweitige Korrelationen, Skalierungsgesetze mit universellen Exponenten usw. Es handelt sich hier aber nicht um Systeme, wie man sie sonst aus der Statistischen Physik kennt, denn es gibt keine Wahrscheinlichkeits–Beschreibung, die durch einen Temperaturparameter gekennzeichnet wäre. Man spricht deshalb von "geometrischen" Pha-

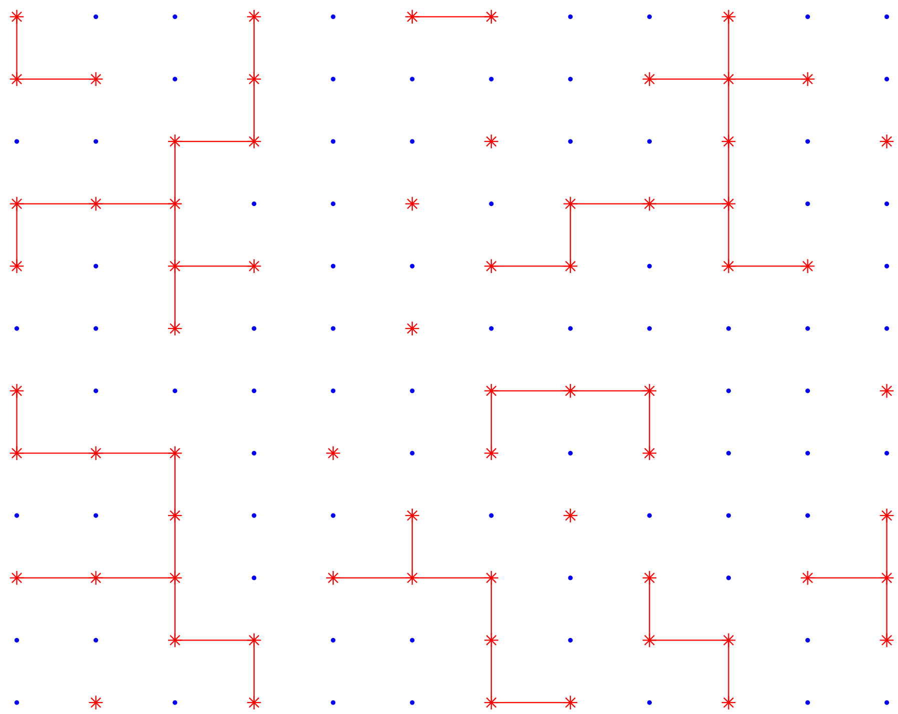

Abbildung 16: Zufällige Gitterbelegung zur Punktperkolation mit $p = 0.5$.

senübergängen.

6.2 Einstieg in die Numerik

Das folgende Hauptprogramm belegt die Punkte eines quadratischen Gitters der Größe $L \times L$ mit Wahrscheinlichkeit p. Wir nehmen freie Randbedingungen, es gibt keine (äusseren) Nachbarn zu den Randpunkten, insbesondere wird *nicht periodisch* geschlossen.

Zunächst wird über die Besetzung entschieden: `feld(x,y) = -1` markiert leere und `feld(x,y) = 0` besetzte Punkte. In einem Plot werden die Gitterpunkte mit entsprechend verschiedene Symbolen belegt und besetzte Punkte verbunden. Ein Output ist in Fig.16 zu sehen. Dann werden zwei verschiedene Analyse–Programme gerufen, die die "0"–Einträge in `feld` mit je einer Cluster–Nummerierung überschreiben. Die beiden Verfahren werden in den nächsten Abschnitten besprochen. Hier ist das Hauptprogramm punkt_perk.m

```
% file punkt_perk.m
% Hauptprogramm zur Perkolation

L=12;  % Gittergroesse L*L
p=0.5;  % Besetzungswahrscheinlichkeit
rng(0);           % Standard initialisierung der Zufallszahlen
disp('Punktbesetzung:');
feld=-(rand(L,L) > p);
flipud(feld.')  % druck-plot gegen (x,y)
% Bedeutung feld-Werte: -1=leer, 0=besetzt, andere:
% Cluster Indizes

% Plot
figure();axis([0 L+1 0 L+1]);hold on;
% besetzt/leer:
for x=1:L, for y=1:L
   if feld(x,y) >= 0, plot(x,y,'*r'); else plot(x,y,'.b'); end
end,end
% Linien zwischen besetzten Nachbarn
for x=1:L-1, for y=1:L
    if feld(x,y) >= 0 && feld(x+1,y) >= 0, plot([x x+1],[y y],'-r'); end
end,end
for x=1:L, for y=1:L-1
    if feld(x,y) >= 0 && feld(x,y+1) >= 0, plot([x x],[y y+1],'-r'); end
end,end
axis off;

pause
feld1=feld;

disp('Ergebnis, Baumsuche:');
[feld] = baum_analyse(feld);
flipud(feld.')

disp('Ergebnis, Hoshen-Kopelman:');
[feld1] = hoshen_kopelman(feld1);
```

```
flipud(feld1.')
```

Der erste Teil des Output des obigen Skript sieht so aus:

```
punkt_perk
Punktbesetzung:

ans =

 0    -1    -1     0    -1     0     0    -1    -1     0    -1    -1
 0     0    -1     0    -1    -1    -1    -1     0     0     0    -1
-1    -1     0     0    -1    -1     0    -1    -1     0    -1     0
 0     0     0    -1    -1     0    -1     0     0     0    -1    -1
 0    -1     0     0    -1    -1     0     0    -1     0     0    -1
-1    -1     0    -1    -1     0    -1    -1    -1    -1    -1    -1
 0    -1    -1    -1    -1    -1     0     0     0    -1    -1     0
 0     0     0    -1     0    -1     0    -1     0    -1    -1    -1
-1    -1     0    -1    -1     0    -1     0    -1    -1    -1     0
 0     0     0    -1     0     0     0    -1     0    -1     0     0
-1    -1     0     0    -1    -1     0    -1     0     0    -1     0
-1     0    -1     0    -1    -1     0     0    -1     0    -1    -1

Ergebnis, Baumsuche:

ans =

16    -1    -1    10    -1    17    17    -1    -1    12    -1    -1
16    16    -1    10    -1    -1    -1    -1    12    12    12    -1
-1    -1    10    10    -1    -1    14    -1    -1    12    -1    15
10    10    10    -1    -1    13    -1    12    12    12    -1    -1
10    -1    10    10    -1    -1    12    12    -1    12    12    -1
-1    -1    10    -1    -1    11    -1    -1    -1    -1    -1    -1
 2    -1    -1    -1    -1    -1     8     8     8    -1    -1     9
 2     2     2    -1     7    -1     8    -1     8    -1    -1    -1
-1    -1     2    -1    -1     3    -1     6    -1    -1    -1     5
 2     2     2    -1     3     3     3    -1     4    -1     5     5
-1    -1     2     2    -1    -1     3    -1     4     4    -1     5
-1     1    -1     2    -1    -1     3     3    -1     4    -1    -1
```

Als erstes sehen wir die Belegung der Gitterpunkte mit 0 und -1 und danach das Resultat der ersten Clusteranalyse durch Baumsuche. Die Nullen sind durch Indizes ersetzt, die jeweils ein Cluster kennzeichnen wie es das Auge in Fig.16 unmittelbar erkennt. Es wurden insgesamt 16 Cluster gefunden, die verschiedene Anzahlen von Punkte umfassen, teilweise sogar nur einen einzigen. Ihre Größenverteilung etc. kann man nach dieser Identifizierung nun 'maschinell' untersuchen.

6.3 Clusterkonstruktion durch Baumsuche

Nun wollen wir verstehen, wie die eindeutige Identifizierung durchgeführt wurde mit

```
function [feldr] = baum_analyse(feld)
% Clusteranalyse mit Baumsuche
feldr=feld; % return Variable
[Lx,Ly]=size(feld);

liste=zeros(Lx*Ly,2); % Arbeitsliste fuer Clustersuche
cluster=0;

for ya=1:Ly
for xa=1:Lx
   if feldr(xa,ya)==0
       cluster=cluster+1; % neuer Index vergeben
       feldr(xa,ya)=cluster;
       liste(1,:)=[xa ya];
       n=1; % letzter benutzter Listenplatz
       i=0;
       while i < n
           i=i+1;
           x=liste(i,1);y=liste(i,2);
           % rechten Nachbarn pruefen
           if x < Lx,
               if feldr(x+1,y)==0
                   feldr(x+1,y)=cluster;
                   n=n+1;
                   liste(n,:)=[x+1 y];
```

```
                end
            end
            % linken Nachbarn pruefen
            if x > 1
                if feldr(x-1,y)==0
                    feldr(x-1,y)=cluster;
                    n=n+1;
                    liste(n,:)=[x-1 y];
                end
            end
            % oberen Nachbarn pruefen
            if y < Ly
                if feldr(x,y+1)==0
                    feldr(x,y+1)=cluster;
                    n=n+1;
                    liste(n,:)=[x y+1];
                end
            end
            % unteren Nachbarn pruefen
            if y > 1
                if feldr(x,y-1)==0
                    feldr(x,y-1)=cluster;
                    n=n+1;
                    liste(n,:)=[x y-1];
                end
            end
        end % while loop: Cluster komplett
%       fprintf('\n Cluster % i aus %i  Punkten \n',cluster,n)
   end % erster if loop
end %loop xa
end %loop ya
```

Wir laufen über alle Gitterpunkte in der x-y Ebene. Findet man eine Null in `feldr` (Kopie des übergebenen `feld`), so ist dieser Punkt belegt und gehört noch keinem bereits mit einem Index gekennzeichneten Cluster an. Man wählt einen neuen noch nicht vergebenen Index und heftet ihn diesem Punkt und allen durch belegete Nachbarpaare im Sinne der Punktperkolation mit ihm verbundenen Punkten an. Der Name des Verfahrens stammt daher, dass man jeden Cluster sofort komplett konstruiert, indem man bis in alle Verästelungen (→ “Baum”) seiner Struktur vordringt.

Zur Programmierung: Wir verwenden das Hilfsfeld `liste` um zwischendurch die Koordinaten von Punkten abzulegen, deren Nachbarn noch untersucht werden müssen. Um die Funktion wirklich zu verstehen, muss man das Programm für eine (kleine) Verteilung wohl mal auf Papier ‘zu Fuß’ abarbeiten.

6.4 Clusterzerlegung nach Hoshen–Kopelman

Die Routine sieht to aus:

```
function [feldr] = hoshen_kopelman(feld)
% Clusteranalyse a la Hoshen Kopelman
feldr=feld; % return Variable
[Lx,Ly]=size(feld);

label=zeros(Lx*Ly+1,1); % Arbeitsliste
LEER=Lx*Ly+1;           % groesser als der maximale Clusterindex
label(LEER)=LEER;       % auf sich selbst zeigend
neu=1;

for y=1:Ly
for x=1:Lx
    if feldr(x,y) == -1
        feldr(x,y)=LEER; % leerer Punkt, umbenannt
    else                 % besetzter Punkt::
        links=LEER;
        if x > 1, links=feldr(x-1,y); end
        unten=LEER;
        if y > 1, unten=feldr(x,y-1);
            while label(unten) < unten, unten=label(unten);end
        end
        if links==LEER && unten==LEER % neuen Index vergeben
            feldr(x,y)=neu;
            label(neu)=neu;          % auf sich selbst zeigend
            neu=neu+1;
```

```
        elseif links < unten
            feldr(x,y)=links;
            if unten < LEER, label(unten)=links; end % Cluster verbinden
        else   % unten <= links
            feldr(x,y)=unten;
            if links < LEER, label(links)=unten; end % Cluster verbinden
        end
    end
end % loop x
end % loop y
% vorlaeufige Nummerierung fertig

% alle auf gute label setzen, Leerstellen wieder -1 ::
for y=1:Ly
for x=1:Lx
    n=feldr(x,y);
    while label(n) < n, n=label(n);end % gutes label finden
    feldr(x,y)=n;
    if n == LEER, feldr(x,y)=-1;end % Leerstellen -> -1
end
end
```

Auch hier beginnt man unten links[13] bei $x = y = 0$, geht aber dann systematisch zeilenweise von unten nach oben vor und betrachtet nur Nachbarn links und unten (falls vorhanden). Wieder vergibt man bei jedem neuen (möglicherweise nur Fragment eines) Clusters eine neue Nummer. Nun trifft man aber schon in der zweiten Reihe unseres Beispiels auf das Problem, die beiden bislang als "2" und "5" nummerierten Fragmente verbinden zu müssen:

```
-1    -1     5     ?
-1     1    -1     2    -1    -1     3     3    -1     4    -1    -1
```

Man geht nun so vor, daß man dem Verbindungspunkt die kleinere der beiden Zahlen zuweist und in einem zusätzlichen Indexfeld vermerkt, daß "5" ein Teil von "2" ist: `label(5) = 2`. Das ergibt vorläufig

. . . .

[13] Man beachte die operation `flipud(feld.')` in `punkt_perk.m` um so zu drucken.

```
 8     8     2    -1     9     9     3    -1     4    -1    10     7
-1    -1     5     2    -1    -1     3    -1     6     4    -1     7
-1     1    -1     2    -1    -1     3     3    -1     4    -1    -1
```

Auch "6,8,9,10", gelten fortan als "schlecht", die anderen Indizes, die (noch) nicht an ein anderes Clusterfragment angeschlossen wurden, erkennt man an `label(i) = i` und nennt sie (vorläufig) "gute" Indizes. Es kann auch vorkommen, daß die Abbildung `label` erst nach mehreren Schritten auf eine gute Zahl führt. Um die Verschachtelung nicht zu tief werden zu lassen, hat es sich bewährt, neue Punkte nicht direkt mit den Zahlen der Nachbarpunkte zu verbinden, sondern, wenn diese schlecht sind, sich erst zum guten Index durchzufragen. (Das betrifft nur den unteren Nachbarn, der linke ist immer gut indiziert, weil er ja gerade vorher erst gesetzt wurde.)

In einem (einfachen) weiteren Durchlauf werden schließlich alle besetzten Punkte mit ihrem (endgültig) guten Index belegt.

Die Funktion `hoshen_kopelman()` implementiert dieses Verfahren. Es sei noch angemerkt, daß die unbesetzten Punkte hier nicht mit -1, sondern vorübergehend mit $L^2 + 1$ markiert werden, das ist größer als alle Clusterindizes und erleichtert das Setzen neuer Label auf das Minimum der Nachbarn. Der Output sieht nun insgesamt so aus:

```
Ergebnis, Hoshen-Kopelman:

ans =

25    -1    -1    17    -1    27    27    -1    -1    20    -1    -1
25    25    -1    17    -1    -1    -1    -1    20    20    20    -1
-1    -1    17    17    -1    -1    23    -1    -1    20    -1    24
17    17    17    -1    -1    22    -1    20    20    20    -1    -1
17    -1    17    17    -1    -1    20    20    -1    20    20    -1
-1    -1    17    -1    -1    18    -1    -1    -1    -1    -1    -1
 2    -1    -1    -1    -1    -1    14    14    14    -1    -1    16
 2     2     2    -1    13    -1    14    -1    14    -1    -1    -1
-1    -1     2    -1    -1     3    -1    11    -1    -1    -1     7
 2     2     2    -1     3     3     3    -1     4    -1     7     7
-1    -1     2     2    -1    -1     3    -1     4     4    -1     7
-1     1    -1     2    -1    -1     3     3    -1     4    -1    -1
```

Man erkennt (natürlich) dieselbe Clusterstruktur wie zuvor, nur ist die Nummerierung jetzt anders — die Indexmanipulationen haben Lücken hinterlassen.

Das Hoshen–Kopelman–Verfahren hat gegenüber der obigen Baumsuche den Vorteil, die Systempunkte nicht in unregelmäßiger Reihenfolge, sondern in zwei systematischen Durchgängen zu bearbeiten, der Datenzugriff ist deshalb schneller. Das war insbesondere von Bedeutung, als man sehr große Systeme noch nicht im Hauptspeicher unterbringen konnte (sondern z.B. nur auf Magnetbändern!). Auch bei der Parallelisierung sehr großer Gitter könnte dies vorteilhaft sein. Es sind weniger Anfragen an die Nachbarpunkte nötig, weil man jeden Bond nur in einer Richtung ansieht. Ausserdem ist es denkbar, die Clusteranalyse in Subvolumina unabhängig durchzuführen (z. B. auf verschiedenen Prozessoren) und diese Cluster dann an den Grenzflächen a la Hoshen-Kopelman "zusammenzunähen".

Ein kleiner Nachteil ist, daß kompliziertere geometrische Eigenschaften der Cluster (z.B. die lineare Ausdehnung) nicht unmittelbar ablesbar, sondern nur über eine weitere Durchmusterung zu erhalten sind.

Was den Rechenaufwand in Abhängigkeit von der Systemgröße anbelangt, so ist bei der Baumsuche klar, daß er proportional zum Volumen ist, denn jeder Punkt wird nur einmal aus jeder Richtung berührt (und einmal bei der Suche nach einem neuen Clusteranfang). Beim Hoshen–Kopelman–Verfahren sind dagegen pathologische Fälle möglich, wo die wachsende Tiefe der Index–Rekursionen einen Aufwand verursacht, der schneller wächst als das Volumen. Das scheint aber in realistischen Anwendungen nicht zu passieren, sondern nur bei speziell "präparierten" Belegungen. Beide Methoden lassen sich ohne weiteres auf mehr als zwei Dimensionen erweitern.

6.5 Charakteristische Größen der Clusterzerlegung

Am wichtigsten ist natürlich die Frage, ob in einem gegebenen System mit Besetzungswahrscheinlichkeit p Perkolation auftritt, anschaulich: ob ein Cluster das "gesamte" System durchdringt. Wenn dies für $p \geq p_c$ der Fall ist, wird p_c als *Perkolationsschwelle* oder *kritischer Punkt* bezeichnet. Wie kann man das genauer definieren?

Als Maß für die *Größe* eines einzelnen Clusters nimmt man die Anzahl s seiner Punkte ($s \geq 1$). Um weiterhin korrekte Größen zu definieren, verschieben wir den Übergang zum unendlichen System auf später und nehmen die Anzahl der Gitterpunkte zunächst als endlich an. Die mittlere Anzahl *pro*

Gitterpunkt der Cluster mit Größe s wird mit n_s bezeichnet. Damit wird bei Punktperkolation (im Mittel)

$$\sum_s sn_s = p, \tag{6.1}$$

denn das ist gerade die Wahrscheinlichkeit dafür, daß ein Punkt zu irgendeinem Cluster gehört, also besetzt ist. Wenn wir nun zufällig einen besetzten Punkt auswählen, dann werden wir mit Wahrscheinlichkeit $sn_s/\sum_{s'} s'n_{s'}$ finden, daß er zu einem Cluster der Größe s gehört, und so können wir eine *mittlere Clustergröße*

$$S(p) = \frac{\sum_s s^2 n_s}{\sum_s sn_s} \tag{6.2}$$

definieren.

Nun kann die *Perkolationsschwelle* p_c definiert werden als

$$\lim_{L\to\infty} S(p) = \infty \quad \text{für} \quad p \geq p_c, \tag{6.3}$$

denn der "unendliche" Cluster läßt S divergieren. Hier ist es die Erfahrung mit solchen Systemen, die besagt, dass es typisch *einen* cluster gibt, der mit wachsendem L 'mitwächst'.

Um Größen definieren zu können, die auch für $p \geq p_c$ einen Grenzwert bei $L \to \infty$ haben, geht man folgendermaßen vor: man definiert einen *perkolierenden* oder *überbrückenden* Cluster (spanning cluster) durch die Eigenschaft, daß er entweder
(i) zwei gegenüberliegende Ränder verbindet, oder
(ii) alle Ränder miteinander verbindet.
Die letztere Bedingung ist offenbar stärker. (i) ist leichter zu verifizieren und wird deshalb meist benutzt, hat aber den Nachteil, daß es prinzipiell mehrere solche Cluster nebeneinander geben könnte; die Wahrscheinlichkeit dafür ist aber z.B. in einem großen, quadratischen System winzig klein.

Nun definiert man endgültig

$$S(p) = \frac{\sum_s' s^2 n_s}{\sum_s' sn_s}, \tag{6.4}$$

wobei in $\sum_s'$ der Beitrag des perkolierenden Clusters ggf. wegzulassen ist. Diese Definition ist auch bei $L \to \infty$ sinnvoll und gibt allgemein die mittlere Größe der "endlichen" Cluster an.

Ferner sei $P_\infty(p)$ die Wahrscheinlichkeit dafür, daß ein besetzter Punkt zum perkolierenden Cluster gehört (wenn er existiert, sonst $P_\infty = 0$). Im unendlichen System erwartet man $P_\infty(p) = 0$ für $p < p_c$ und ein Anwachsen $P_\infty(p) \to 1$ bei $p \to 1$. Hier handelt es sich also um den *Ordnungsparameter* der Perkolation (vergleichbar z.B. mit der Magnetisierung eines Ferromagneten ober– und unterhalb der Curie–Temperatur).

Die Clustergröße war oben durch die Anzahl der Punkte definiert, was keine direkte Aussage über die räumliche Ausdehnung darstellt. Deshalb steht die Einführung einer Längenskala noch aus. Dazu können wir vom "Trägheitsradius" R_s des Clusters s ausgehen: es werde von den Punkten $\vec{x}_i, i = 1 \ldots s$ gebildet, dann liegt sein "Schwerpunkt" bei

$$\vec{X} = \frac{1}{s} \sum_i \vec{x}_i, \tag{6.5}$$

es wird

$$R_s^2 = \frac{1}{s} \sum_i (\vec{x}_i - \vec{X})^2, \tag{6.6}$$

und durch Mittelung über alle *endlichen* Cluster entsteht die *Korrelationslänge*

$$\xi(p) = \left(\overline{R_s^2}\right)^{1/2}. \tag{6.7}$$

Eine andere mögliche Definition zieht die *Korrelationsfunktion* $G(r)$ heran, das ist die Wahrscheinlichkeit dafür, daß zwei Punkte im Abstand r zum selben Cluster gehören. An der Asymptotik kann man ξ ablesen:

$$G(r) \sim \exp(-r/\xi) \quad r \to \infty. \tag{6.8}$$

Die Übereinstimmung beider Definitionen kann man zeigen, es erfordert aber einigen einigen Aufwand.

6.6 Exakte Lösung des eindimensionalen Problems

Wie wir sehen werden, gibt es in einer eindimensionalen Kette bei Punkt– oder Bondwahrscheinlichkeit $p < 1$ keine Perkolation, da immer eine endliche Wahrscheinlichkeit $1-p$ reicht, die Verbindung zu unterbrechen. Trotz dieser Pathologie ist es nützlich, die Perkolation in einer Dimension zu diskutieren, denn man findet leicht exakte Ausdrücke für viele interessante Größen.

Wir betrachten beispielsweise Punktperkolation in einer linearen Kette der Länge $L \to \infty$. Cluster kann man hier ohne Mühe als Blöcke besetzter Punkte identifizieren, die durch leere Punkte getrennt sind.

Die Wahrscheinlichkeit dafür, daß ein bestimmter Abschnitt der Kette ein Cluster der Größe s bildet, beträgt $p^s(1-p)^2$. Wir haben einen Faktor p für jeden der besetzten Clusterpunkte und je einen $(1-p)$ für die notwendig angrenzenden leeren Punkte. Dieser Cluster kann L verschiedene Positionen einnehmen, wobei wir Randeffekte der (relativen) Größe s/L im Hinblick auf den Limes $L \to \infty$ vernachlässigen. Wenn wir im gleichen Sinne die Überlappung vernachlässigen, dann gibt es im Mittel $Lp^s(1-p)^2$ s–Cluster im System der Größe L und demnach ist

$$n_s(p) = p^s(1-p)^2. \tag{6.9}$$

Die Relation (6.1) kann man leicht nachrechnen:

$$\begin{aligned}\sum_s sn_s(p) &= (1-p)^2\sum_s sp^s \\ &= (1-p)^2 p\partial_p \sum_s p^s \\ &= (1-p)^2 p\partial_p \frac{p}{1-p} \\ &= p.\end{aligned}$$

Mit demselben Trick erhält man

$$\begin{aligned}\sum_s s^2 n_s(p) &= (1-p)^2(p\partial_p)^2\sum_s p^s \\ &= p\frac{1+p}{1-p}\end{aligned}$$

und damit

$$S(p) = \frac{1+p}{1-p}. \tag{6.10}$$

Man erkennt $S(p) = \infty$ bei $p = 1$, d.h.

$$p_c = 1. \tag{6.11}$$

Die Korrelationslänge bekommt man mit folgender Überlegung: damit zwei Punkte im Abstand r zum selben Cluster gehören, müssen sie mit allen dazwischen liegenden Punkten besetzt sein, was eine Wahrscheinlichkeit p^{r+1} hat. Die Korrelationsfunktion verhält sich also wie

$$G(r) \sim \exp(-r/\xi) \tag{6.12}$$

mit

$$\xi = -\frac{1}{\log p}. \tag{6.13}$$

6.7 Skalengesetze im unendlichen System

Beim Durchgang durch den Perkolationspunkt verhalten sich die Observablen des Systems — zu $L \to \infty$ extrapoliert — in auffälliger Weise, nämlich nicht–analytisch. Der Ordnungsparameter P_∞, der zuvor identisch null war, beginnt zu wachsen, die mittlere Clustergröße S und die Korrelationslänge ξ durchlaufen eine Singularität. Man versucht häufig, so auch bei der Perkolation (und bei anderen Phasenübergängen höherer Ordnung), dieses Verhalten durch Skalengesetze mit *kritischen Exponenten* zu parametrisieren:

$$P_\infty \sim (p - p_c)^\beta \tag{6.14}$$

$$S(p) \sim |p - p_c|^{-\gamma} \tag{6.15}$$

$$\xi(p) \sim |p - p_c|^{-\nu} \tag{6.16}$$

Die Theorie dieser kritischen Phänomene ist schwierig: Stichwort "Renormierungsgruppe". In der Praxis helfen oft nur numerische Simulationen um die suggerierte Form zu bestätigen und Werte für die Exponenten zu finden.

In einer Dimension haben wir explizite Lösungen: an (6.10) liest man $\gamma = 1$ ab. Aus (6.12) folgt wegen $-\log p \simeq (1-p)$ bei $p \to p_c = 1$: $\nu = 1$. β ist hier wegen $p_c = 1$ nicht (direkt) bestimmbar.

Für Punktperkolation in zwei Dimensionen sind die Exponenten auch exakt bekannt:

$$\beta = 5/36 \tag{6.17}$$

$$\gamma = 43/18 \tag{6.18}$$

$$\nu = 4/3 \tag{6.19}$$

$$\tag{6.20}$$

Allgemein gilt in d Dimensionen die Relation

$$2\beta + \gamma = \nu d \tag{6.21}$$

(hyperscaling).

6.8 Skalierung mit der Systemgröße

Die Bestimmung der kritischen Exponenten aus numerischen Simulationen ist schwierig, weil in den endlichen Systemen das obige Verhalten nur gilt, solange $\xi \ll L$, d.h. in gebührender Entfernung vom Perkolationspunkt. Wenn man bei endlichem L durch $p = p_c$ hindurchläuft, erscheinen die Singularitäten "gerundet".

Man kann aber aus der Not eine Tugend machen, direkt *am kritischen Punkt* arbeiten (wo $\xi = \infty$ wäre im unendlichen Volumen) und die Systemgröße als einzig relevante Längenskala ansehen. Bei Vergrößerung von L ergeben sich dann neue Skalengesetze (*finite size scaling*, FSS), deren Exponenten mit denen des *unendlichen* Systems *in der Umgebung* von p_c zusammenhängen.

Ein einfaches (nur heuristisches) Argument soll zeigen, wie die neuen Ansätze aussehen: In einer kleinen Umgebung des Perkolationspunkts, wo $\xi > L$ ist (im ∞ System), übernimmt L die Rolle der charakteristischen Länge, und man setzt

$$\xi \approx L \sim |p - p_c|^{-\nu}. \tag{6.22}$$

Also ist $|p-p_c|$ der Abstand vom kritischen Punkt zu dem man gehen müsste, damit ξ auf L sinkt. Man eliminert damit $(p - p_c)$ aus den Skalengesetzen der anderen Observablen:

$$P_\infty(p_c) \sim L^{-\beta/\nu} \tag{6.23}$$

$$S(p_c) \sim L^{\gamma/\nu} \tag{6.24}$$

als asymptotisches Verhalten bei $L \to \infty$ *am kritischen Punkt*. Aus Messungen von $P_\infty(p_c)$ und $S(p_c)$ an Systemen von wachsendem (aber endlichem!) L kann man also die Verhältnisse von Exponenten β/ν bzw. γ/ν gewinnen.

Dies ist für numerische Simulationen sehr praktisch. Erneut ist es gelungen,"Finite Size Effekte" durch Theorie (die wir hier aber nur berichtet haben) mit Physik zu verbinden. Die Bestimmung von β/ν und γ/ν am Perkolationspunkt des zweidimensionalen quadratischen Gitters ($p_c = 0.5927$) wird in einer Übungsaufgabe behandelt.

7 Monte Carlo Integration

Bei vielen bedeutenden numerischen Verfahren spielen Zufallszahlen eine große Rolle. Das ist gerade dort der Fall, wo ein System zu komplex ist, um seine beschreibenden Gleichungen, selbst wenn diese bekannt sind, für einen allgemeinen Fall zu lösen. Oft ist es dann dennoch möglich, ein System, das diesen Gleichungen folgt, zu simulieren und wenigstens approximative Teilaussagen zu erhalten indem man typische Trajektorien oder Konfigurationen betrachtet. Die Perkolation war ein solches Beispiel: Wo wir die Wahrscheinlichkeiten n_s nicht berechnen konnten, haben wir (endlich) viele Ensembles, die sich gemäß n_s verhalten, erzeugt und untersucht. Auch in vielen anderen Bereichen insbesondere der statistischen Physik geht es darum, hochdimensionale Integrale oder Summen über einen Phasenraum auszuwerten. Hier erlauben Monte Carlo (MC) Verfahren einen recht allgemeinen Zugang, und damit wollen wir uns befassen.

7.1 Ideale Zufallszahlen

Im Folgenden werden wir Verfahren studieren, die als Input Zufallszahlen $\eta_i, i = 1, 2, \ldots$ benötigen, die in einem bestimmten Wertebereich liegen und reell sind. Man stelle sich zur "Ziehung" einer Zufallszahl also vor, eine Funktion (Generator) aufzurufen, die als Antwort ohne irgendeine Eingabe einen zufälligen Wert η ausspuckt. Dabei ist die Verteilung $p(\eta) \geq 0$ per Konstruktion bekannt. Sie legt die Wahrscheinlichkeit

$$W(\eta \in [a, b]) = \int_a^b \mathrm{d}u\, p(u) \tag{7.1}$$

fest, daß bei einer beliebigen Ziehung eine Zufallszahl im Intervall $[a, b]$ liegt. Weiter hat p über den gesamten Wertebereich das Integral Eins, bezieht sich also auf eine Ziehung. Das geläufigste Beispiel sind flache Zufallszahlen im Intervall $[0, 1]$ mit $p(\eta) = 1$.

Stochastische Verfahren benötigen viele Zufallszahlen und wir betrachten Folgen $\eta_1, \eta_2, \ldots, \eta_N$, die durch wiederholtes Aufrufen des Generators entstehen. Eine wichtige Eigenschaft, die für die meisten Anwendungen essentiell ist, ist die Unabhängigkeit dieser Zahlen. D. h. die Wahrscheinlichkeit, daß ein solches N-Tupel als Vektor betrachtet in einem N-dimensionalen infinitesimalen Volumen $\mathrm{d}V$ um den Vektor $(u_1, u_2, \ldots, u_N)$ liegt ist gegeben durch

$p(u_1)p(u_2)\cdots p(u_N)\mathrm{d}V$. Insbesondere ist sie auch unabhängig davon, an welcher Stelle die N Zahlen einer längeren Gesamtfolge entnommen werden.

Die Einhaltung der vorgegebenen Verteilung p und die statistische Unabhängigkeit einander folgender Zahlen sind die Qualtitätsmerkmale guter Zufallszahlen Generatoren. In der Realität ist insbesondere die Unabhängigkeit nur näherungsweise erfüllt. Die verbleibenden kleinen Korrelationen aufeinanderfolgender Zahlen führen dann bei Verfahren, die Unabhängigkeit voraussetzten, zu entprechenden systematischen Fehlern. Übliche Generatoren, die ihre Zahlen ja auch schnell liefern sollen, sind deterministisch, weshalb man genauer auch von Pseudozufallszahlen spricht. Sie bestehen aus einem dynamischen System, das nach irgendeiner Initialisierung bei jedem Aufruf nach irgendeinem Algorithmus um einen Schritt evolviert. Solche Systeme ähneln denen, die man beim Studium von Chaos betrachtet. Chaos in Phasenraum bedeutete ja gerade ein (exponentiell) rasches "Vergessen" der Anfangsbedingung, ist also hier erstrebenswert.

Eine besonders drastische (aber primitive) Korrelation stellt die Periodizität dar: dann wiederholt sich die Folge nach M Aufrufen, ist also völlig durch die vorherigen Zahlen festgelegt. Jeder endliche Generator ist periodisch. Es gibt dann nämlich irgendwelche k Bits, die den Zustand des Systems kodieren. Dann muß nach spätestens 2^k Schritten derselbe Zustand wieder vorliegen und das ganze sich wiederholen. Es ist allerdings nicht schwer, die Periodizität astronomisch groß zu machen. Das größere Problem ist es, subtilere Arten von Korrelationen zu vermeiden. Wir setzen im folgenden voraus, hinreichend gute Zufallszahlen von MATLAB `rand, randn` zu bekommen, so dass wir ihre Defekte vernachlässigen können. Bezüglich der verwendeten Algorithmen für Zufallszahlen hat sich MATLAB über die Versionen weiterentwickelt (insbesondere auch die Art der Initialisierung) und es lohnt sich, einmal das Kapitel `rand` der aktuellen Dokumentation zu lesen um eine Idee von der Bedeutung der Qualität von Zufallszahlen und dem deshalb getriebenen Aufwand zu bekommen.

7.2 Monte Carlo Integration

Die Monte Carlo Integrationsmethode ist interessant bei hochdimensionalen Integralen, wie wir sehen werden.

Wenn wir eine Funktion f über erzeugte Zufallszahlen mitteln,

$$\langle f \rangle = \frac{1}{N} \sum_{i=1}^{N} f(\eta_i), \tag{7.2}$$

so ist klar, daß gilt

$$\lim_{N\to\infty} \langle f \rangle = \int \mathrm{d}u \; p(u) f(u) =: \bar{f}. \tag{7.3}$$

Das Integral geht über den Wertebereich der Zufallszahlen. Die entscheidende Frage ist die nach der Konvergenz der Schätzung mit N. Dazu stellen wir uns vor, daß wir die ganze Schätzung M mal durchführen ($M \to \infty$) mit Zufallszahlen $\eta_{i,a}, i = 1, \ldots, N,\ a = 1, \ldots, M$, die alle natürlich unabhängig sind. Dann haben wir die mittlere Abweichung

$$\begin{aligned} \sigma(N,f)^2 &= \frac{1}{M} \sum_{a=1}^{M} \left[\langle f \rangle - \bar{f} \right]^2 \qquad (7.4) \\ &= \frac{1}{N^2} \sum_{i,j=1}^{N} \frac{1}{M} \sum_{a=1}^{M} [f(\eta_{i,a}) - \bar{f}][f(\eta_{j,a}) - \bar{f}] \end{aligned}$$

Nun gilt

$$\lim_{M\to\infty} \frac{1}{M} \sum_{a=1}^{M} [f(\eta_{i,a}) - \bar{f}][f(\eta_{j,a}) - \bar{f}] = \delta_{ij} \mathrm{var}(f) \tag{7.5}$$

mit

$$\mathrm{var}(f) = \int \mathrm{d}u \; p(u) \left[f(u) - \bar{f} \right]^2 \tag{7.6}$$

und somit

$$\sigma(N,f) = \sqrt{\frac{\mathrm{var}(f)}{N}}. \tag{7.7}$$

Damit kann man den erwarteten Fehler der Schätzung angeben,

$$\langle f \rangle = \bar{f} \pm \sqrt{\frac{\mathrm{var}(f)}{N}}. \tag{7.8}$$

Für $i = j$ ist (7.5) einfach eine Anwendung von (7.2). Für $i \neq j$ wird M mal jeweils ein Zweitupel $(\eta_{i,a}, \eta_{j,a})$ von unabhängigen Zufallszahlen gezogen, die mit der Produktverteilung $p(u_1)p(u_2)$ (Dichte) in einer kleinen Fläche um den Vektor (u_1, u_2) liegen. Im Limes $M \to \infty$ ensteht

$$\left(\int \mathrm{d}u\, p(u) f(u) - \bar{f}\right)^2 = 0.$$

Zur Fehlerabschätzung benötigen wir $\mathrm{var}(f)$, und diese Größe ist normalerweise genausowenig bekannt wie $\bar{f}$. Der Ausweg ist, $\mathrm{var}(f)$ gleichzeitig ebenfalls durch MC Integration zu berechnen. Dazu verwendet man

$$\mathrm{var}(f) = \int \mathrm{d}u\, p(u) f^2(u) - \bar{f}^2 \approx \langle f^2 \rangle - \langle f \rangle^2. \tag{7.9}$$

Selbstverständlich ist diese Schätzung wieder mit einem Fehler $\propto 1/\sqrt{N}$ behaftet. Dieser "Fehler des Fehlers" wird normalerweise vernachlässigt; ansonsten müßte man das Ganze noch eine Stufe höher treiben.

Wenn die Schätzwerte $E = \langle f \rangle$ um $\bar{f}$ herum normalverteilt sind, $\propto \exp[-(E - \bar{f})^2/(2\sigma^2)]$, was für nicht zu pathologische f und nicht zu kleine N zu erwarten ist[14], so hat σ mehr quantitative Bedeutung: 68 % aller Schätzungen liegen im Intervall $[\bar{f} - \sigma, \bar{f} + \sigma]$.

Die Konvergenz der MC Integration geht also proportional zu $1/\sqrt{N}$, wo N ja auch die Anzahl der Funktionsberechnungen $f(\eta)$ ist entsprechend der Zahl der Stützstellen. Vergleichen wir mit einem Standard Integrationsverfahren. Für ein Volumen L^D wähle man Stützstellen mit Raster h in jeder Dimension, und der Algorithmus konvergiere proportional zu $(h/L)^n = N^{-n/D}$ mit der Anzahl Stützstellen $N = (L/h)^D$. Für kleine D, insbesondere D=1, kann man den Exponenten n/D leicht größer als $1/2$ machen, so daß Standardverfahren wie Simpson mit $n = 4$ viel effektiver sind. Für jeden gegebenen Algorithmus der Ordnung n ist aber MC in hohen Dimensionen $D > 2n$ asymptotisch im Vorteil. Ebenso simpel wie verblüffend. Es ist damit oft die einzige Möglichkeit. Dennoch muß man stets zur Verdopplung der Genauigkeit den Rechenzeit Etat vervierfachen, was manchmal frustriert.

Es kommt natürlich auch noch auf den Vorfaktor $\mathrm{var}(f)$ an. Dieser verschwindet genau dann, wenn f konstant ist. Die Werte sind dann unabängig von η, und *eine* "Schätzung" genügt. Diese Betrachtung ist nicht so leer wie es scheint. Angenommen, wir wollen

$$I = \int \mathrm{d}u\, h(u) \tag{7.10}$$

---

[14]Dies ist der Inhalt des zentralen Grenzwertsatzes.

schätzen mit p-verteilten Zufallszahlen. Dann müssen wir in der obigen Betrachtung

$$f = \frac{h}{p} \tag{7.11}$$

wählen, und konstantes f bedeutet, daß die Verteilung p bis auf einen Normierungsfaktor dem Integranden h entspricht,

$$p(u) = \frac{1}{I} h(u). \tag{7.12}$$

Um also genau p-verteilte η zu produzieren, wird man I kennen müssen! Praktisch wird man aber ein p wählen, das man gut erzeugen kann, und das möglichst viel von der Variation von h erfaßt im Sinne minimaler Varianz von f. Diese Methode heißt Importance Sampling. Je besser dies gelingt, desto schneller konvergiert die MC Schätzung. Man hat prinzipiell jedoch auch immer mit flachen Zufallszahlen einen korrekten Algorithmus, aber die nötigen N könnten unpraktikabel werden. Diese Gefahr besteht besonders bei oszillierenden h, da ja p als Wahrscheinlichkeit nur positiv sein kann.

7.3 Beispiel für Monte Carlo Integration

Als Beispiel wollen wir nun das Integral

$$I = \int_0^1 \mathrm{d}u \, \frac{1}{1+u^2} = \arctan(u)\Big|_0^1 = \pi/4 \tag{7.13}$$

per MC berechnen [10]. Dazu nehmen wir flach verteilte Zufallszahlen, die in MATLAB unter **rand** zur Verfügung stehen. Die Kernzeilen des MATLAB Programms sind

```
absc=[1:N];
x=rand(1,N);    % N Zufallszahlen
f=1./(1+x.*x); % Integrand (Vektor)
f=cumsum(f);    % kumulative Summe
f=f./absc;      % Mittelwerte
```

Mit **cumsum** wird aus einem Vektor ein neuer gebildet, der die Teilsummen enthält (cumulative sum). Damit wurden für $N = 200$ Abb. 17 und 18 erzeugt. Wenn man solche Experimente wiederholt, findet man durchaus auch

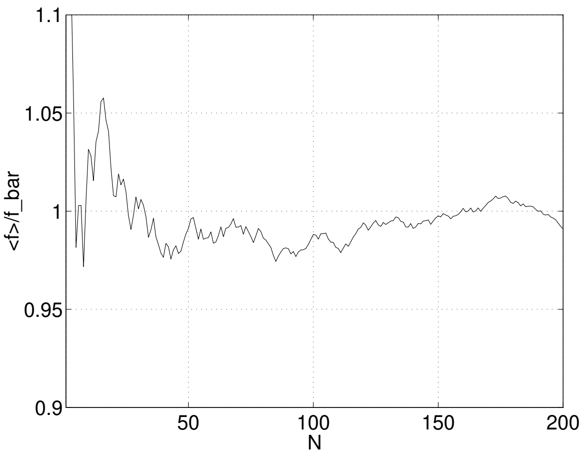

Abbildung 17: $\langle f \rangle / I$ für *eine* MC Schätzung

Fälle, wo mehr oder weniger die ganze Fieberkurve außerhalb des "Konvergenzkorridors" liegt, wenn auch niemals um große Faktoren. Das zeigt, daß die Fehlerabschätzung nur ungefähr gilt, und daß 2σ Abweichungen durchaus praktisch vorkommen. Auch muß man beim Betrachten der Abbildungen bedenken, daß für wachsendes N die früheren MC-Schätzungen weiter im Mittel drin sind. Die verschiedenen Teile der Kurve sind also nicht etwa unabhängig. Liegt sie einmal unter dem exakten Wert, so wird sie eine Weile dort bleiben.

In Abb. 19 sehen wir die Verteilung von 1000 unabhängigen MC Schätzungen mit N=200. Die Varianz ist für diesen Integranden ebenfalls exakt bekannt,

$$\begin{aligned} \mathrm{var}(f) &= \frac{1}{4} + \frac{\pi}{8} - \frac{\pi^2}{16} \approx 0.0258 \\ \sqrt{\frac{\mathrm{var}(f)}{200}} &\approx 0.011, \end{aligned} \tag{7.14}$$

was gut zu der Breite der Verteilung paßt.

Wir wollen nun für dieses einfache Beispiel anders verteilte Zufallszahlen verwenden. Wenden wir auf einen Typ Zufallszahlen eine monotone Funktion b an,

$$\eta \to \tilde{\eta} = b(\eta), \tag{7.15}$$

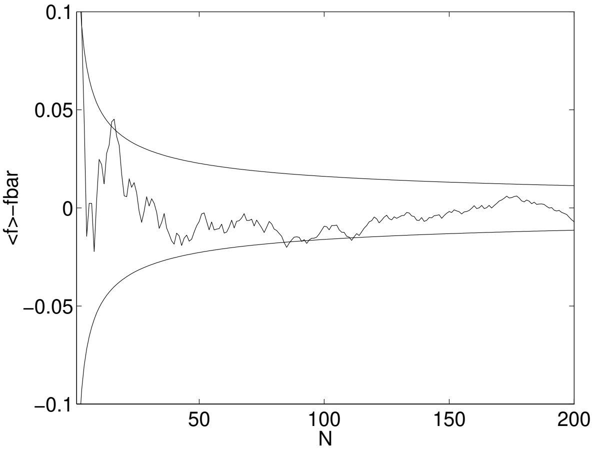

Abbildung 18: $\langle f\rangle - I$ und “Konvergenzkorridor” $\pm\sigma(N, f)$

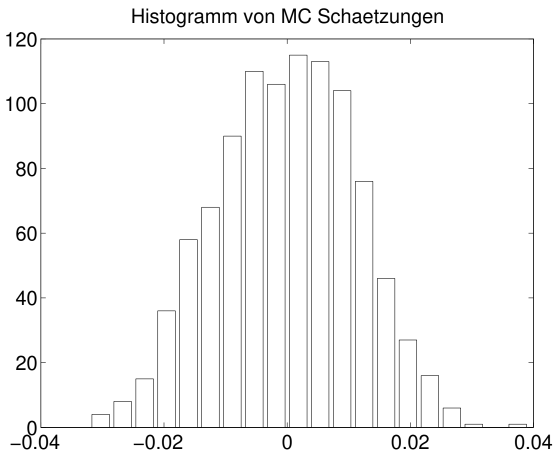

Abbildung 19: Verteilung der Abweichung von 1000 Berechnungen $\langle f\rangle$ mit N=200

so erhalten diese die Verteilung

$$\tilde{p}(\tilde{\eta})\mathrm{d}\tilde{\eta} = p(\eta)\mathrm{d}\eta = p(B(\tilde{\eta}))B'(\tilde{\eta})\mathrm{d}\tilde{\eta}, \tag{7.16}$$

wobei B die Umkehrfunktion von b ist. Der Bereich der $\tilde{\eta}$ ist natürlich der Bildbereich der ursprünglichen Zahlen unter b. Ausgehend von $p = 1$ würde man für optimales Importance Sampling anstreben, daß $\tilde{p}(u)(1+u^2)$ konstant wird,

$$B'(u) \stackrel{!}{=} c\frac{1}{1+u^2}. \tag{7.17}$$

Zusammen mit $B(0) = 0, B(1) = 1$, führt dies auf

$$B = \arctan,\ b = \tan,\ c = 4/\pi, \tag{7.18}$$

und nun ist trivialerweise der konstante Integrand $1/c$ zu integrieren.

Realistischer ist es, daß man keine exakte Konstanz erreicht, aber eine Verteilung findet, die ähnlich ist. Hier wäre z. B. eine lineare Verteilungsfunktion denkbar, die wie der Integrand monoton fällt und $p(0) = 2p(1)$ hat,

$$p(u) = \frac{2}{3}(2-u). \tag{7.19}$$

Sie entsteht durch eine Transformation mit

$$b(\eta) = 2 - \sqrt{4 - 3\eta} \tag{7.20}$$

aus flachen Zufallszahlen. Der Integrand ist dann gegeben durch

$$f(u) = \frac{1}{p(u)(1+u^2)} \tag{7.21}$$

mit der Varianz[15]

$$\mathrm{var}(f) = \int_0^1 \mathrm{d}u\ p(u)f^2(u) - (\pi/4)^2 = \frac{36\log(2) + 90 + 42\pi - 25\pi^2}{400} \approx 0.0004. \tag{7.22}$$

Der Fehler ist also bei diesem sampling etwa 8 mal kleiner, obwohl auch (nur) mit $N^{-1/2}$ sinkend.

[15] Dank sei Maple für dieses Integral

7.4 Zufallszahlen in MATLAB

In MATLAB gibt es selbstverständlich eingebaute Zufallszahlen, die wir ja schon verwendet haben. Wie wir schon sahen, erhält man durch den Aufruf der Funktion **rand** flach verteilte Zufallszahlen zwischen 0 und 1. Nach jedem Start von MATLAB erhält man die gleiche Folge, die einer Standard Initialisierung der Variablen entspricht, die den Zustand des Generators beinhalten. Will man diesen Zustand abspeichern, z. B. um mit Hilfe dieser Information später mit der Zufallsfolge fortzufahren, so geht das wie folgt:

```
>> s=rng

s =

     Type: 'twister'
     Seed: 0
    State: [625x1 uint32]
```

Die Rückgabevariable der Funktion `rng` (ohne Argument) ist hier vom Typ `structure`. Insbesondere bekommt man, wenn man `s.State` eingibt, einen Vektor von 625 ganzen Zahlen, die den Zustand des Generators beinhalten. Wenn man `s` in eine Datei schreibt (`save`) und irgendwann wieder einliest (`load`, dann `rng(s)`), dann geht die Folge von Zufallszahlen genau an der Stelle weiter, wo man `s` ausgelesen hat. Will man verschiedene Folgen haben ohne ein `s` zur Verfügung zu haben, das aus dem Generator stammt, so kann man auch mit `rng(j,'twister')` initialisieren, wobei `j` eine ganze Zahl ist. `j=0` entspricht der Standard Initialisierung bei Programmstart. Die Funktion `rng` erlaubt viele weitere Argumente, und man kann sogar eine ganze Batterie verschiedener Generatoralgorithmen wählen, die dann bei Aufrufen von `rand` arbeiten an Stelle des Defaults mit Namen 'twister'.

8 Monte Carlo Simulation im Ising Modell

In diesem Abschnitt befassen wir uns mit der Monte Carlo Simulation von statistischen Systemen am Beispiel des Ising Modells. Monte Carlo Simulation ist eine zentrale Methode in der statistischen und Festkörper Physik und in einem Zweig der theoretischen Elementarteilchen Physik, der Gitter Feldtheorie. Sie spielt eine zentrale Rolle im Forschungsprogramm der Arbeitsgruppe COM (Wolff) und bei unseren Kollaborationspartnern bei DESY-Zeuthen (Sommer, Jansen).

8.1 Das Ising Modell auf einem kubischen Gitter

Zunächst soll das Ising Modell definiert werden [11]. Es ist ein einfaches statistisches System und damit ein natürlicher Startpunkt. Eine mögliche Motivation ergibt sich als stark vereinfachtes Modell eines Ferromagneten. Wir stellen uns ein einfach kubisches Kristallgitter vor, das wir allerdings verallgemeinernd nicht nur in 3 sondern in D Dimensionen betrachten wollen. Gitterplätze (lattice sites) sind durch D ganzzahlige Koordinaten gegeben:

$$\text{Gitterplatz} = x = (x_0, x_1, \ldots, x_{D-1}),\ x_\mu \in \mathbb{Z} \tag{8.1}$$

Für thermodynamische Definitionen und natürlich zur Simulation brauchen wir zunächst ein endliches Volumen, obwohl am Ende oft der Grenzübergang zum unendlichen steht. Dazu schränken wir die Koordinaten ein[16],

$$0 \leq x_\mu < L,\ V = L^D, \tag{8.2}$$

wobei hier die Anzahl V der so gegebenen Punkte das Volumen ist. Das bezieht sich auf Gittereinheiten, in denen die Elementarzelle das Volumen Eins besitzt.

Ferromagnetische Effekte in Festkörpern stammen von Elektronen Spins. Im Modell werden diese durch Freiheitsgrade $s(x) \in \{-1, +1\}$ dargestellt, die zwei möglichen Spin Ausrichtungen pro Gitterplatz entsprechen. Diese Diskretheit ist offenbar in der zugrundeliegenden Quantentheorie begründet. Eine Konfiguration $s \,\hat{=}\, \{s(x)\}$ für alle V Gitterplätze spezifiziert nun im Modell genau einen Zustand des Systems; sein Phasenraum besteht also aus 2^V diskreten solchen möglichen Konfigurationen. In der statistischen Physik gibt

[16] Man könnte die Kantenlängen auch richtungsabhängig nehmen

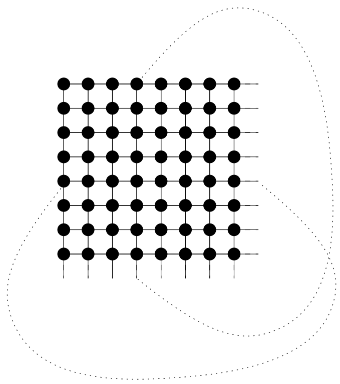

Abbildung 20: Gitter für Ising Modell in $D = 2$.

man die exakte Beschreibung der Dynamik solcher Systeme auf und macht Wahrscheinlichkeitsaussagen. Das Wunder ist, daß dies für den Bereich, wo thermodynamisches Gleichgewicht gilt, auf einfache Weise möglich ist, wenn man nur die Energie H des Systems für jede Konfiguration kennt. Beim Ising Modell lautet diese

$$-H(s) = \epsilon \sum_{\langle xy \rangle} s(x)s(y) + B \sum_x s(x) \tag{8.3}$$

Die erste Summe läuft über alle Kanten (links) im Gitter und $s(x), s(y)$ sind die beiden Spins an den Enden. Der Parameter ϵ, positiv für Ferromagneten, steuert eine energetische Bevorzugung von parallelen gegenüber antiparallelen Spins. Der zweite Term beschreibt den Einfluß eines magnetischen Feldes (reell, skalar, entlang der Quantisierungsachse).

Jeder Punkt im Inneren des Gitters partizipiert offenbar an $2D$ solchen Kanten. Am Rand sind es i. a. weniger, und wir brauchen zusätzliche Definitionen, was zu tun ist. Die Situation ist für zwei Dimensionen in Abb.20 verdeutlicht. Läßt man die am Rand herausragenden Kanten bzw. ihre Energieterme einfach weg, so nennt man dies freie Randbedingungen. Identifiziert man sie, wie mit den punktierten Linien in zwei Fällen angedeutet, alle mit den Punkten am gegenüberliegenden Rand, so hat man periodische Randbedingungen und das Gitter ist ein Torus. Dies ist die Standardwahl. Sie hat eine besonders hohe Symmetrie, da alle Punkte gleichwertig sind. Auch hat man Verschiebungsinvarianz um Vielfache der Kantenlänge.

Wenn wir dies nun noch in den Koordinaten formalisieren, auch im Hinblick auf Programmierung, so sehen wir gleich, daß das Konzept sich problemlos auf D Dimensionen verallgemeinern läßt. Man führt Einheitsvektoren für die Gitterrichtungen ein,

$$\hat{\mu} = (0, \ldots, 0, \underbrace{1}_{\mu-\text{Komponente}}, 0, \ldots, 0),\ 0 \leq \mu \leq D-1. \tag{8.4}$$

Wenn man nun die Addition von Torus Koordinaten immer modulo L versteht, so kann man den ersten Energieterm in (8.3) als

$$\sum_{x,\mu} s(x)s(x+\hat{\mu}) \tag{8.5}$$

schreiben. Die Summe geht nun also über $V \times D$ völlig gleichwertige Kanten.

Zum Programmieren mit Feldern auf dem Torus, ist es sicher naheliegend jeweils Felder mit D Indizes für jede Komponente einzuführen. Natürlicher und einfacher (= weniger Fehler) wird es jedoch, insbesondere bei Programmierung für allgemeine D, wie folgt. Wir nummerieren alle Punkte x eindeutig durch ganze Zahlen,

$$n_x = x_0 + x_1 L + x_2 L^2 + \ldots + x_{D-1} L^{D-1},\ 0 \leq n_x < V \tag{8.6}$$

und deklarieren als Konfiguration für alle D ein Feld mit nur einem Index. Zur Berechnung der Indizes der $2D$ Nachbarn zu n_x ist einige Rechnung erforderlich. Diese führt man nur einmal durch und speichert die Nachbarindizes in einem ganzzahligen Feld mit $V \times 2D$ Elementen. Dieses erlaubt es ohne neue Rechnung zu beliebigen Nachbarn zu "hüpfen". Auf diese Weise gelangt man zu kompakten Programmen, in denen man D tatsächlich als veränderbaren Parameter hat. Selbstverständlich ist diese Strategie recht großzügig mit Speicherplatz, was auf modernen Rechnern aber kein Problem ist.

8.2 Observable im statistischen Gleichgewicht

Es ist ein Resultat der statistischen Mechanik bzw. Thermodynamik, daß man unter bestimmten Bedingungen eine korrekte und vorhersagekräftige statistische Beschreibung der Physik erhält, wenn man postuliert, daß ein System mit einer Wahrscheinlichkeit proportional zu $\exp(-\beta H)$ in allen seinen Zuständen sein kann (kanonisches Ensemble). Hier ist β ein Parameter der das Ensemble vollständig festlegt und der inversen Temperatur entspricht. Speziell für das Ising Modell ist der Erwartungswert einer beliebigen Funktion der Konfigurationen (Observable) $\mathcal{A}(s)$ gegeben durch

$$\langle \mathcal{A} \rangle = \frac{1}{Z} \sum_s \mathcal{A}(s)\, \mathrm{e}^{-\beta H(s)} \tag{8.7}$$

mit der normierenden Zustandssumme

$$Z = \sum_s \mathrm{e}^{-\beta H}. \tag{8.8}$$

Beide Summen gehen über alle 2^V möglichen Konfigurationen der V Spins.

Eine offensichtliche mögliche Observable ist die mittlere (innere) Energie,

$$E = \langle H \rangle = -\frac{\partial \ln Z}{\partial \beta}. \tag{8.9}$$

Durch wiederholtes Differenzieren nach dem Feld bekommt man Potenzen der mittleren Magnetisierung wie z. B.

$$\begin{aligned} M &= \sum_x s(x) \\ \langle M \rangle &= \frac{1}{\beta} \frac{\partial \ln Z}{\partial B} \end{aligned} \tag{8.10}$$

Neben diesen aus der Energie abgeleiteten Obervablen, erhält man detaillierte Informationen aus n-Punkt Korrelationsfunktionen (jedes x^i ist ein Gitterplatz)

$$G^{(n)}(x^1, x^2, \ldots, x^n) = \left\langle s(x^1) s(x^2) \cdots s(x^n) \right\rangle. \tag{8.11}$$

Speziell für die Zweipunktfunktion schreiben wir

$$G(x, y) = \langle s(x) s(y) \rangle. \tag{8.12}$$

Sie beschreibt die Korrelation zwischen zwei möglicherweise weit entfernten Spins. Ist sie z. B. positiv, so gibt es eine Tendenz für diese Spins parallel zu stehen. Wenn dies für beliebig weit entfernte[17] Spins selbst bei $B = 0$ der Fall ist, so liegt spontane Magnetisierung vor, das eigentliche Phänomen Ferromagnetismus. Dies passiert im Ising Modell für $D \geq 2$ und genügend kleine Temperatur (großes β).

[17] Für die Theorie ist dies nach einem Grenzübergang $L \to \infty$ wörtlich gemeint. In der Physik geht es um makroskopische Distanzen (Weiss'sche Bezirke)

8.3 Exakte Lösung in $D = 1$

In nur einer Dimension ist das Problem wie so oft stark vereinfacht und analytisch lösbar. Die Spins bilden einfach eine Kette, die im Falle periodischer Randbedingungen zu einem Ring zusammengebogen ist. Nur der Einfachheit halber spezialisieren wir uns hier auf $B = 0$ und wählen die Einheit der Temperatur so, daß $\epsilon = 1$ gilt. Wir definieren die Transfermatrix durch ihre Matrixelemente

$$T_{\sigma\sigma'} = \exp(\beta\sigma\sigma'). \tag{8.13}$$

Zeilen und Spalten werden hier durch zwei Spins $\sigma, \sigma' = -1, +1$ abgezählt. Dann können wir im periodischen Fall schreiben[18]

$$Z = \sum_{s(0),s(1),\ldots,s(L-1)} T_{s(0)s(1)} T_{s(1)s(2)} \cdots T_{s(L-1)s(0)} = \mathrm{tr}\left[T^L\right]. \tag{8.14}$$

Die Eigenwerte von T sind

$$\lambda_\pm = \exp(\beta) \pm \exp(-\beta) \tag{8.15}$$

mit Eigenvektoren

$$\psi_\pm = \frac{1}{\sqrt{2}}(\pm 1, 1). \tag{8.16}$$

Somit ist

$$Z = \lambda_+^L + \lambda_-^L \approx (2\cosh(\beta))^L, \tag{8.17}$$

wobei die Näherung für hinreichend große Systeme $L \gg \exp(2\beta)$ gilt und im Folgenden angenommen wird. Damit ist die Energie

$$E = -L\tanh(\beta). \tag{8.18}$$

Die Korrelationsfunktion zwischen entfernteren Spins bei $x < y$ ist für $L \gg y - x$ gegeben durch

$$\begin{aligned} \langle s(x)s(y)\rangle &= \frac{\mathrm{tr}\left[ST^{y-x}ST^{L-y+x}\right]}{\mathrm{tr}[T^L]} \\ &= \frac{\left(\lambda_-^{y-x}\lambda_+^{L-y+x}\right)}{\lambda_+^L} = \tanh(\beta)^{y-x}, \end{aligned} \tag{8.19}$$

[18] In einer Dimension gibt es nur eine Komponente $x = x_0$

wobei S analog zu T aber diagonal ist mit Elementen

$$S_{\sigma\sigma'} = \sigma\,\delta_{\sigma\sigma'}, \tag{8.20}$$

und daher

$$S\psi_\pm = \psi_\mp \tag{8.21}$$

benutzt werden konnte. Die Korrelationen fallen in $D = 1$ also für jede Temperatur exponentiell zu Null ab, es gibt keine spontane (ohne B) Magnetisierung. Dies ist ein allgemeines Resultat in einer Dimension und nicht auf das Ising Modell beschränkt. Die Korrelation von nächsten Nachbarn lautet

$$G(x, x + \hat{\mu}) = \tanh(\beta) \tag{8.22}$$

und ist wegen der Symmetrie des Torus ortsunabhängig. Daher trägt zur Energie jeder link gleich bei und senkt die Energie monton mit β wachsend ab, vgl. (8.18).

Eine exakte Lösung ist auch in $D = 2$ bekannt, wenngleich sie ungleich schwieriger zu bekommen ist. Eine Möglichkeit hierzu ist wieder die Diagonalisierung der Transfermatrix, die allerdings nun 2^L dimensional ist. Dieses Modell mit seiner bekannten Onsager Lösung [11], die einen Phasenübergang zur magnetisierten Phase aufweist, ist sozusagen der "harmonische Oszillator der statistischen Physik". Jedes Konzept wir dort getestet.

8.4 Monte Carlo Simulation am Beispiel Ising Modell

Nun sollen Mittelwerte von Obervablen numerisch berechnet werden. Vollständiges Summieren ist für interessante Fälle indiskutabel. Die meisten Beiträge wären aber bedeutungslos wegen der Unterdrückung durch den Boltzmann Faktor $\exp(-\beta H)$. Hier kommen nun Monte Carlo Verfahren mit importance sampling, die wir bei der MC Integration kennengelernt hatten, zum Tragen.

Der dominante Faktor im Summanden ist offenbar der Boltzmann Faktor, der für alle Konfiguration zwischen $\exp(\pm\beta DV)$ enorm variieren kann. Typische Observable sind beschränkter, z. B. nur ± 1 bei Korrelationen. Daneben kommt es aber auch auf die Anzahl der existierenden Konfiguration mit vergleichbaren Boltzmann Faktoren an (Entropie). Eine gute Idee wäre offenbar, beliebige Konfigurationen zu produzieren mit der Wahrscheinlichkeit

$$P(s) = \frac{1}{Z}\,\mathrm{e}^{-\beta H(s)}. \tag{8.23}$$

Ist $s^1, s^2, \ldots, s^N$ eine (lange) so verteilte Folge, so gilt wie bei der MC Integration

$$\langle \mathcal{A}(s) \rangle = \frac{1}{N} \sum_{i=1}^{N} \mathcal{A}(s^i) \left(1 + \mathrm{O}(N^{-1/2})\right). \tag{8.24}$$

Für Systeme wie das Ising Modell gibt es keine praktikablen direkten Algorithmen die *unabhängige* korrekt verteilte s (independent sampling) produzieren. Statt dessen gibt es Monte Carlo Algorithmen, die die Konfiguration s^i benutzen und modifizieren in s^{i+1}:

$$\ldots s^i \overset{MC}{\to} s^{i+1} \overset{MC}{\to} s^{i+2} \ldots$$

Wenn s^{i+1} nur von s^i abhängt und einige zusätzliche Bedingungen erfüllt sind, spricht man von einer Markov Kette (Markov chain). Diese Verfahren können so kontruiert werden, daß bei einer langen Folge asymptotisch die gewünschte Verteilung mit dem Boltzmann Faktor eintritt. Allerdings ist hierbei s^{i+1} eben *nicht unabhängig* von s^i. Gegenüber der früher diskutierten einfachen Monte Carlo Integration hat dies erhebliche Konsequenzen für die erwarteten statistischen Fehler und ihre Schätzung. Außerdem muß man von irgendeiner Konfiguration s^1 starten. Korrekte Monte Carlo Algorithmen führen beweisbar von jeder beliebigen Startkonfiguration asymptotisch zur Boltzmann Verteilung, brauchen dazu aber eine mehr oder weniger große Zahl von Schritten. Der Beginn der Folge wird vom willkürlichen Start abhängen und somit für die Mittelung besser weggelassen. Wieviel, das hängt vom Algorithmus ab und muß untersucht werden. Man nennt in Analogie zur Thermodynamik dieses "Einschwingen" auf typische Konfigurationen Thermalisierung (equilibration).

Der Übergang von einer Konfiguration zur nächsten erfolgt zufällig und ist charakterisiert durch vorgegebene Übergangswahrscheinlichkeiten $W(s^i \to s^{i+1})$, gemäß denen der Prozess unter Benutzung von Zufallszahlen abläuft. Als Wahrscheinlichkeiten erfüllen sie

$$W(s \to s') \geq 0, \ \sum_{s'} W(s \to s') = 1 \tag{8.25}$$

Damit W einen Algorithmus mit den oben postulierten Eigenschaften ergibt, muß gelten

$$\sum_{s} P(s)\, W(s \to s') = P(s') \qquad \text{Stabilität} \tag{8.26}$$

$$W^n(s \to s') > 0 \ \forall s, s' \qquad \text{Ergodizität} \tag{8.27}$$

mit irgendeinem n. Hier ist W^n die Übergangswahrscheinlichkeit bei Hintereinanderausführung entsprechend der Matrix Potenzierung von W. Die zweite Bedingung besagt, daß man in n Schritten prinzipiell von jedem s zu jedem s' gelangen kann. Das ist u. a. notwendig um von jedem s^1 in die relevanten Bereiche des Phasenraumes zu gelangen. Zur Interpretation der ersten Bedingung stellen wir uns ein großes Ensemble von Markov Ketten vor. Wenn nun die i-ten Konfigurationen dieser vielen Prozesse mit Häufigkeit $P(s)$ verteilt sind, dann gewährleistet die Stabilität von P unter W, daß dies auch für die s^{i+1} gilt. P ist also (Links)eigenvektor von W mit Eigenwert 1. Ein mathematischer Satz für positive normierte Matrizen wie W besagt nun, daß die Beträge aller anderen Eigenwerte echt kleiner sind als 1. Daher führt wiederholte Anwendung von W (MC Iterationen) auf die Verteilung P, andere Komponenten "sterben aus". Die Geschwindigkeit der Thermalisierung hängt vom Abstand zwischen 1 und dem nächstkleineren Eigenwert ab, mit dem die führende unerwünschte Komponente evolviert.

8.5 Lokale Monte Carlo Algorithmen

Wie wir sehen werden, sind viele W konstruierbar, die einem prinzipiell richtigen Algorithmus entsprechen. Es wird sich weiter zeigen, daß Verfahren effektiv sind, wenn sie für einen bestimmten Rechenaufwand von einander möglichst unabhängige s^i produzieren. Zunächst aber müssen die mit W zulässigen Übergänge überhaupt realisierbar sein. Das ist nur der Fall, wenn $W(s \to s') \neq 0$ nur für wenige s' gilt, denn zwischen diesen muß ja gewählt werden. Dies — weniger die Effektivität — ist sehr allgemein erfüllt für lokale MC Verfahren mit Elementarschritten (x fest, beliebig)

$$W_x(s \to s') = \left(\prod_{y \neq x} \delta_{s(y)s'(y)}\right) \; w_x(s(x) \to s'(x)). \tag{8.28}$$

Es wird also nur die Änderung des einen Spins bei x zugelassen, und somit können nur zwei Zustände erreicht werden, $s' = s^{x,+}$ und $s' = s^{x,-}$, wo in der Konfiguration $s^{x,\pm}$ jeweils an der einen modifizierten Stelle $s(x)$ durch ± 1 ersetzt ist und der Rest hier ungeändert bleibt. w_x ist nun eine 2×2 Matrix. Aus (8.25) und (8.26) folgt nun leicht, daß

$$\frac{w_x(+ \to -)}{w_x(- \to +)} = \frac{P(s^{x,-})}{P(s^{x,+})} \tag{8.29}$$

gelten muß.

Diese Bedingung wird standardmäßig mit dem Wärmebad oder mit dem Metropolis Algorithmus erfüllt. Beim Wärmebad wird $\sigma = s'(x)$ unabhängig vom alten Spin $s(x)$ gewählt

$$\begin{aligned} w_x(s(x) \to \sigma) &= \frac{P(s^{x,\sigma})}{P(s^{x,+}) + P(s^{x,-})} \\ &= \frac{\exp[\sigma\beta b(x)]}{\exp[+\beta b(x)] + \exp[-\beta b(x)]} \end{aligned} \tag{8.30}$$

Im letzten Schritt ist $b(x)$ das lokale Magnetfeld

$$b(x) = B + \sum_{y=\text{Nachbarn}(x)} s(y). \tag{8.31}$$

Wichtig ist, daß die Wahl eines neuen Spins bei x (update) nur lokale Operationen benötigt, deren Anzahl nicht mit dem Volumen anwächst. Dies liegt an der lokalen Wechselwirkung in $H(s)$. Alle übrigen Energieterme heben sich weg in (8.30).

Bei Metropolis ist der Gesichtspunkt, daß eine Änderung $s \to \tilde{s} = s^{x,-s(x)}$ *vorgeschlagen* wird (Spinflip bei x). Dieser Vorschlag wird akzeptiert mit der Wahrscheinlichkeit

$$\begin{aligned} w_x(s(x) \to -s(x)) &= \min(1, P(\tilde{s})/P(s)) \\ &= \min(1, \exp(-\beta[H(\tilde{s}) - H(s)])) \\ &= \min(1, \exp(-2\beta s(x) b(x))), \end{aligned} \tag{8.32}$$

sonst bleibt es bei s. In beiden Fällen ist leicht nachzuweisen, daß (8.29) gilt.

Ergodizität wird erreicht, wenn man nacheinander lokale Updates bei allen x durchführt,

$$W_{\text{sw}} = W_{x^1} W_{x^2} \cdots W_{x^V}. \tag{8.33}$$

Hier ist $\{x^1, x^2, \ldots, x^V\}$ eine Anordnung der Sites in beliebiger Reihenfolge. Dabei ist die triviale Tatsache von Bedeutung, daß (8.25) und (8.26) für Produkte gelten, wenn sie für die Faktoren gelten. W_{sw} hängt durchaus von der Reihenfolge der Faktoren ab, da sie für benachbarte x nicht kommutieren. Verschiedene Reihenfolgen liefern i. a. verschiedene korrekte Algorithmen. Diese können sich bezüglich der Unabhängigkeit der s^i durchaus unterscheiden. Ein solcher kompletter Durchgang durch das Gitter heißt Sweep. Er ist ergodisch und erfordert O(V) Operationen. Man könnte auch ein site x zum update zufällig wählen (random site updating). Dann hätte man

$$W_{\rm rs} = \frac{1}{V} \sum_x W_x, \tag{8.34}$$

und dies alleine wäre ergodisch (mit $n = V$). Praktisch hat sich dieses Verfahren als nicht vorteilhaft erwiesen. Es liegt wohl daran, daß nach V Schritten (Aufwand etwa wie ein Sweep) einige sites typischerweise kein update bekommen, andere mehrere.

Viele Algorithmen benutzen das detaillierte Gleichgeweicht (detailed balance) als hinreichende (nicht notwendige) Bedingung für Stabilität,

$$P(s)W(s \to s') = P(s')W(s' \to s). \tag{8.35}$$

Stabilität folgt durch Summation über s oder s'. Diese Form hatte auch die Bedingung an das lokale w_x (8.29), so daß detailed balance in diesem Spezialfall auch notwendig war. Dies liegt an den nur zwei Werten der Ising Spins, ist i. a. aber nicht der Fall. Die Hintereinanderausführung von Schritten mit detailed balance liefert i. a. für das Produkt nur noch Stabilität. Alle hier durchgeführten Betrachtungen lassen sich ziemlich direkt verallgemeinern, auf Spins mit mehr und sogar kontinuierlichen Werten, über die dann an jedem site integriert wird.

8.6 Autokorrelation und statistische Fehler

MC Schätzungen a_i für eine Observable A werden aus den mit irgendeinem Algorithmus sukkzessive im Rechner produzierten Konfigurationen s^i abgeleitet,

$$a_i = \mathcal{A}(s^i),\ i = 1, \ldots, N, \quad A = \langle \mathcal{A}(s) \rangle. \tag{8.36}$$

Diese aufeinanderfolgenden Schätzwerte sind nicht unabhängig, was die Fehlerabschätzung verkompliziert. Dabei wollen wir hier annehmen, daß die Folge von Schätzwerten erst nach der Thermalisierung beginnt. Bevor Konfiguration s^1 auftritt, wurden also schon so viele Iterationen durchgeführt, daß das System thermalisiert ist und der willkürliche Anfangszustand in beliebig guter Näherung "vergessen" wurde. Dann ist s^1 also bereits eine "typische" Gleichgewichtskonfiguration. Wir werden später einen Hinweis bekommen, wieviele Iterationen eine solche Thermalisierung praktisch benötigt.

Zur Fehlerabschätzung wollen wir wie bei der einfachen MC Integration die mittlere quadratische Abweichung vom exakten Mittelwert bilden,

$$\sigma(N,A)^2 = \left\langle \left(\frac{1}{N} \sum_{i=1}^{N} a_i - A \right)^2 \right\rangle_{\rm MC} . \tag{8.37}$$

Auch hier gilt der zentrale Grenzwertsatz, so dass A, σ die Verteilung unserer Schätzung vollständig festlegen. Die Mittelung $\langle \cdots \rangle_{\rm MC}$ bezieht sich hier auf (im Limes unendlich) viele gleichartige MC Schätzungen (insbesondere aus je N Konnfigurationen), während wir mit $\langle \cdots \rangle$ weiter das statistische Mittel im Sinn von (8.7) meinen. Für das MC-Mittel kann man sich entweder wieder ein Ensemble von Markov Prozessen im Gleichgewicht vorstellen, oder aber einen sehr langen solchen, aus dem man viele Teilfolgen der Länge N herausgreift, die weit genug voneinander entfernt sind, um als unabhängig gelten zu können.

Durch Ausmultiplizieren ergibt sich

$$\sigma(N,A)^2 = \frac{1}{N^2} \sum_{i,j=1}^{N} \Gamma_A(i-j), \tag{8.38}$$

wobei wir die Autokorrelation

$$\Gamma_A(i-j) = \langle (a_i - A)(a_j - A) \rangle_{\rm MC} = \Gamma_A(j-i) \tag{8.39}$$

eingeführt haben. Die Notation impliziert nochmals, daß diese Funktion nur vom Abstand zwischen i und j abhängt und nicht mehr vom Abstand zum Beginn der Folge. Die Autokorrelation bei $i = j$ ist die Varianz,

$$\Gamma_A(0) = \langle (a_i - A)^2 \rangle_{\rm MC} = \langle (\mathcal{A}(s) - A)^2 \rangle = {\rm var}(A). \tag{8.40}$$

Hier wurde benutzt, daß im Gleichgewicht die Konfiguration s^i im MC mit P verteilt ist und so die MC-Mittelung in die statistische des Ising Modells übergeht. Man kann diesen Fall auch statisch nennen, während Mittelungen, die sich auf $i \neq j$ beziehen, dynamisch sind und vom Algorithmus W abhängen.

Wären aufeinanderfolgende Schätzungen unabhängig, so gälte

$$\Gamma_A(i-j) = \Gamma_A(0)\, \delta_{ij} \leftrightarrow \text{unabhängig} \tag{8.41}$$

und damit

$$\sigma(N,A)^2 = \frac{\mathrm{var}(A)}{N} \tag{8.42}$$

in völliger Analogie zur MC Integration.

Üblicherweise würde mit einer solchen Formel der Fehler unterschätzt. Tatsächlich ist Γ_A typischerweise positiv und klingt asymptotisch mit dem Abstand exponentiell ab,

$$\Gamma_A(t) \stackrel{t\to\infty}{\propto} \exp(-t/\tau), \tag{8.43}$$

aber τ kann viele Sweeps lang sein. Für eine brauchbare MC Rechnung benötigt man auf jeden Fall $N \gg \tau$, damit man viele praktisch unabhängige Schätzungen hat[19]. Dann gilt näherungsweise

$$\sum_{i,j=1}^{N} \Gamma_A(i-j)/\Gamma_A(0) \approx N \sum_{t=-\infty}^{\infty} \Gamma_A(t)/\Gamma_A(0) =: 2N\tau_{\mathrm{int},A}. \tag{8.44}$$

Eine effektive oder integrierte Autokorrelationszeit $\tau_{\mathrm{int},A}$ wurde hier definiert[20]. Diese fasst den Effekt von Autokorrelationen auf den Fehler von A zusammen,

$$\sigma(N,A)^2 = \frac{\mathrm{var}(A)}{N/2\tau_{\mathrm{int},A}}. \tag{8.45}$$

Interpretation: Statistischer Fehler wie bei $N/2\tau_{\mathrm{int},A}$ effektiven unabhängigen Schätzungen. Im Prinzip könnten negative Korrelationen den Fehler unter den des unabhängigen Falles drücken ($2\tau_{\mathrm{int},A} < 1$), was aber praktisch kaum gezielt realisierbar ist.

Offenbar beziehen sich die eingeführten MC Zeitskalen auf updates mit irgendeinem Algorithmus, z. B. lokale Wärmebad Sweeps. Würde man hingegen immer schon nach sehr kleinen (und damit "billigen") Schritten – z. B. einem lokalen random site update – "messen", so bekäme man viele Schätzungen, großes N. Gewonnen wäre nichts, da die τ und damit der Fehler dennoch groß wären. Im Gegenteil, da Messungen auch "kosten", würde man wahrscheinlich schließen, daß es günstiger wäre seltener zu messen[21].

[19] Für ein statistisches Verfahren braucht man Statistik!

[20] Wenn (8.43) exakt gilt und $\tau \gg 1$, was oft aber nicht immer realistisch ist, dann ist $\tau_{\mathrm{int},A} \approx \tau$

[21] Die Optimierung hängt von dem Kostenverhältnis Messung/update ab

Im Prinzip können $\tau_{\mathrm{int},A}$ für verschiedene A ganz verschiedenen sein, und keine dieser Größen ist ein strenger Indikator für die Anzahl von updates die zur Thermalisierung zu machen sind. Letztlich sind diese dynamischen Größen von den Eigenwerten und -vektoren von W bestimmt. In der Praxis sind die beobachteten $\tau_{\mathrm{int},A}$ von etwa der gleichen Größenordnung, und Thermalisierungen von 20 bis 100 τ_{max} sind selbstkonsistent gerechtfertigt. Meist ist die Thermalisierung billig, und man bleibt hier auf der sicheren Seite. Mögliche Tests bestehen hier auch aus Starts von verschiedenen Anfangskonfigurationen, z. B. total magnetisiert ($s(x) = +1$) und zufällig. Bei genügender Thermalisierung müssen im Rahmen der Fehler kompatible Mittelwerte rauskommen.

Um in einer Simulation praktisch den Fehler zu schätzen ist man darauf angewiesen, neben der Varianz weitere Information über Γ_A numerisch zu bekommen. Eine Schätzung ist gegeben durch

$$\Gamma_A(t) \simeq \frac{1}{N-t} \sum_{i=1}^{N-t} \left(a_i - \frac{1}{N} \sum_{j=1}^{N} a_j \right) \left(a_{i+t} - \frac{1}{N} \sum_{k=1}^{N} a_k \right). \tag{8.46}$$

Auch diese Schätzung selbst hat wieder einen Fehler, den man zumindest bei einfachen Experimenten vernachlässigen muß. Um $\tau_{\mathrm{int},A}$ zu schätzen, sollte man $\Gamma(t)$ nur soweit summieren, bis es klein ist. Es ist nicht sinnvoll, Beiträge aufzusummieren, die nur noch aus "Rauschen" bestehen. Wenn es vom Datenanfall her möglich ist, ist zu empfehlen, während der teuren Simulation alle a_i zu speichern und diese später zu analysieren, also z. B. $\Gamma(t)$ zu bilden und zu plotten.

Eine einfache Analysemethode besteht auch in der Blockbildung, dem binning. Hier werden die ursprünglichen Messungen a_i blockweise vorgemittelt zu b_k,

$$b_k = \frac{1}{B} \sum_{i=1}^{B} a_{(k-1)B+i}, \quad k = 1, \ldots, N_B = [N/B], \tag{8.47}$$

wo N_B die Zahl der (vollständigen) solchen Bins ist. Dies ist auch rekursiv möglich, z. B. mit $B = 2, 4, 8, \ldots$. Die aufeinanderfolgenden b_k sind natürlich auch noch korreliert, aber wenn B einmal einige τ lang ist, so ist dies ein vernachlässigbarer Randeffekt. Der Mittelwert ist immer der gleiche[22]. Der

[22]Exakt gilt das, solange N durch B teilbar ist und keine Messungen wegfallen

Fehler, für nun (näherungsweise) unabhängige "Blockmessungen" numerisch über die Varianz geschätzt, ist[23]

$$\sigma^2 = \frac{1}{N_B(N_B-1)} \sum_{k=1}^{N_B} \left(b_k - \frac{1}{N_B} \sum_{l=1}^{N_B} b_l \right)^2 . \tag{8.48}$$

Er steigt zunächst mit wachsendem B an, und saturiert dann beim korrekten Wert, wenn die Statistik reicht um dies alles zu sehen.

Binning ist eine Standard Methode, wenn es darum geht, Fehler für nichtlineare Funktionen von Observablen zu bestimmen. Ein Beispiel wäre eine Messung der 2-Punkt Funktion $G(x)$ zwischen Spins mit Abstand x

$$G(x) = \langle \mathcal{O}(s;x) \rangle \tag{8.49}$$

$$\mathcal{O}(s;x) = \frac{1}{V} \sum_z s(z)s(z+x). \tag{8.50}$$

Alle Summen, z. B. $z+x$, und Differenzen sind natürlich unter Beachtung der Torus Periodizität auszuführen (komponentenweise mod L). Man schließt in $\mathcal{O}$ die Summe über z ein, um durch Verwendung der Translationsinvarianz Statistik zu gewinnen. Ein typisches Problem wäre nun eine komplizierte Funktion $F[G(x)]$ zu schätzen, z. B. Parameter aus einem Fit der Korrelation an eine theoretische Form. Ein Schätzwert ergibt sich offenbar durch Einsetzen der Ensemble Mittel über $a_i(x) = \mathcal{O}(s^i;x)$ als Argumente von $F[.]$. Was aber wäre der auf Autokorrelationen korrigierte Fehler der Schätzung? Hier ist nun eine Möglichkeit, für jeden der näherungsweise voneinander unabhängigen zugehörigen Bins $b_k(x)$ die F-Berechnung (z. B. Fit) durchzuführen mit Resultaten $F_k = F[b_k(x)]$ und dann den Fehler zu schätzen wie in (8.48), nur mit F_k an Stelle von b_k selbst.

Praktisch ergibt sich hierbei oft das Problem, daß die Bins zu klein werden und daher ihre Mittel zu stark fluktuieren, um F sinnvoll zu berechnen. Da hilft der Trick des Jackknife Binning: statt der Bins b_k verwenden wir ihr Komplement,

$$c_k = \frac{1}{N-B} \left(\sum_i a_i - B b_k \right) . \tag{8.51}$$

[23] Der Nenner $(N_B - 1)$ ergibt sich bei genauer Analyse daraus, daß die Subtraktion von A auch aus den Schätzungen gebildet ist.

Diese Bins sind $N_B - 1$ mal größer, aber natürlich wieder nicht unabhängig, jedoch in einfacher Weise korreliert. Eine kurze Rechnung zeigt, daß sich aus den Jackknife Bins der Fehler schätzen läßt zu

$$\sigma^2 = \frac{N_B - 1}{N_B} \sum_{k=1}^{N_B} \left(c_k - \frac{1}{N_B} \sum_{l=1}^{N_B} c_l \right)^2 . \tag{8.52}$$

Eine analoge Formel gilt nun für den Fehler der F-Schätzung, wobei man aus den Jackknife Bins gewonnene Werte $F_k = F[c_k(x)]$ benutzt.

Man kann die Fehlerabschätzung für F auch durch Berechnung einer Autokorrelationsfunktion analog zu (8.46) durchführen. Wie das im Detail geht ist in [12] beschrieben.

Literatur

[1] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery, Numerical Recipes, Cambridge University Press
Von diesem nützlichen Buch gibt es verschiedene Ausgaben mit Programmen Fortran, C, C++, `matlab` und anderen Sprachen.
Ältere Ausgaben sind sogar online verfügbar auf `www.nr.com`

[2] G. H. Golub und C. F. van Loan, Matrix Computations, Johns Hopkins University Press, 1989

[3] E. Ott, Chaos in Dynamical Systems, Cambridge University Press 1993

[4] H. G. Schuster, Deterministisches Chaos, VCH Verlagsgesellschaft, Weinheim

[5] H. Goldstein, Klassische Mechanik, AULA-Verlag, Wiesbaden

[6] M. Lüscher, "Volume Dependence Of The Energy Spectrum In Massive Quantum Field Theories. 2. Scattering States," Commun. Math. Phys. **105**, 153 (1986).

[7] M. Lüscher, "Selected Topics in Lattice Field Theory," Lectures given at the Summer School "Fields, Strings and Critical Phenomena", Les Houches, France (1988).

[8] M. Lüscher and U. Wolff, "How To Calculate The Elastic Scattering Matrix In Two-Dimensional Quantum Field Theories By Numerical Simulation," Nucl. Phys. B **339**, 222 (1990).

[9] D. Stauffer und A. Aharony, Perkolationstheorie, VCH Verlagsgesellschaft, 1995

[10] S.E.Koonin und D.C.Meredith, Computational Physics, Addison–Wesley

[11] W. Nolting, Grundkurs Theoretische Physik Bd. 6: Statistische Physik
J. Schnakenberg, Algorithmen in der Quantentheorie und Statistischen Physik
beide: Verlag Zimmermann-Neufang, Ulmen

[12] U. Wolff, “Monte Carlo errors with less errors,” Comput. Phys. Commun. **156**, 143 (2004)
`http://xxx.uni-augsburg.de/pdf/hep-lat/0306017`

[13] J. Hertz, A. Krogh und R.G.Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley

[14] I. Montvay und G. Münster, Quantum Fields on a Lattice, (Cambridge Univ. Press, 1994)

[15] M. Creutz, Quarks, Gluons, Lattices, (Cambridge Univ. Press, 1983)

[16] H. J. Rothe, Lattice Gauge Theories: Am Introduction (Wolrd Scientific, 1992)

[17] J. Kogut, An introduction to lattice gauge theory and spin systems, Reviews of Modern Physics 51 (1979) S.659-713

[18] K. Wilson, Confinement of Quarks, Phys. rev. D10 (1974) 2445