

Untersuchung von Quantensystemen mit Hilfe von gitterfeldtheoretischen Methoden

Heinrich v. Campe

Bachelorarbeit in Physik
angefertigt im Helmholtz-Institut für Strahlen- und
Kernphysik

vorgelegt der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität
Bonn

August 2021

DRAFT

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die Zitate kenntlich gemacht habe.

Bonn,
Datum

.....
Unterschrift

- 1. Gutachter: Prof. Dr. Carsten Urbach
- 2. Gutachterin: Prof. Dr. Anne Jones

Inhaltsverzeichnis

1	Einleitung	1
2	Theorie	3
2.1	Vom Pfadintegral zum Metropolis-Algorithmus	3
2.1.1	Feynman	3
2.1.2	Markov	4
2.1.3	Metropolis	5
2.2	Intermezzo: Der harmonische Oszillator als Beispiel	6
2.3	Das SU(2)-Eichfeld auf einem Gitter	9
2.3.1	Yang-Mills	9
2.3.2	Wilson	10
2.3.3	Haar	13
2.3.4	Wilson und Metropolis	13
2.3.5	Das statische Potential und Wilson-Loops	14
3	Methodik und Implementation	17
3.1	SU(2)-Matrizen	17
3.2	Zufällige SU(2)-Matrizen	18
3.3	Implementierung des Algorithmus' sowie der Observablen	18
3.4	Parameter und Ablauf der durchgeführten Simulationen	19
4	Auswertung	23
4.1	Energie und durchschnittliche Plaquette	23
4.2	Vermessen von Wilson-Loops verschiedener Größe	24
4.3	Extrahierung der Werte für das statische Potential	25
4.4	Confinement (?)	25
5	Fazit	27
A	Zusätzliche Graphen	29
B	Wertetabellen	31
	Literatur	33
	Abbildungsverzeichnis	35

DRAFT

Einleitung

Die theoretische Teilchenphysik erfuhr eine Reihe von Durchbrüchen, nachdem Gell-Mann und Zweig 1964 das Konzept von Quarks eingeführt hatten. So benannten sie die „Bausteine“ des in den vorherigen Jahren immer größer gewordenen Teilchenzoos und erklärten insb. die von Gell-Mann und Ne'mann zuerst beschriebenen $SU(3)$ -Multiplets von Baryonen. [1] 1971 führten Gell-Mann und Fritsch die Farbladung der Quarks als weitere Quantenzahl ein, um Zustände aus drei gleichen Quarks mit dem Pauli-Prinzip in Einklang zu bringen. Im darauffolgenden Jahr betrachteten sie dann die Farb-Gruppe ($SU(3)$) als Eichgruppe und legten so die Grundsteine für eine neue Theorie, der Quantenchromodynamik. [1] Eine wichtige Eigenschaft dieser neuen Theorie ist das sog. *Confinement*: Quarks (und auch Gluonen) können nie ungebunden auftreten. Rein physikalisch lässt sich diese Eigenschaft darauf zurückführen, dass Gluonen (die Mediatoren der beschriebenen, sog. starken Wechselwirkung) selbst Farbladungen tragen, also anders als z. B. Photonen auch untereinander interagieren können. [2]

Leider hat die Quantenchromodynamik einen beträchtlichen Nachteil gegenüber der QED: Bei kleinen Energien lässt sie sich nicht perturbativ betrachten. Hierdurch sind analytische Betrachtungen in diesem Bereich sehr schwierig, weswegen andere Methoden vonnöten sind. Eine solche Alternative bieten eine gitterfeldtheoretische Formulierung, welche erstmals 1974 von Wilson vorgelegt wurde. [2] Dabei präsentierte er gleichzeitig eine erste Motivation für Confinement, welche fünf Jahre später numerisch bestätigt werden konnte.

In dieser Arbeit soll diese numerische Betrachtung nachvollzogen werden: Der Einfachheit halber wird hier nur die $SU(2)$ -Eichsymmetrie betrachtet: Die Simulation des dazugehörigen Eichfelds bietet die Möglichkeit, den Wert des statischen Potentials eines „Quark“-Anti„quark“-Paars für verschiedene Abstände zu berechnen. Confinement sollte dann dadurch zu erkennen sein, dass das Potential (neben einem Coulomb-artigen Verhalten für kleine Abstände) für große Abstände *linear* ansteigt. Entsprechend bleibt die Kraft zwischen „Quark“ und Anti„quark“ konstant, wodurch man sie nicht voneinander isolieren kann.

Folgender Aufbau wurde für die folgenden Abschnitte gewählt: In Kapitel 2 wird erst die allgemeine numerische Methode vorgestellt, welche die numerische Approximation von Feynmans Pfadintegralen ermöglicht. Darauf folgt dann die theoretische Anwendung auf die $SU(2)$ -Eichsymmetrie – von einer allgemeinen Formulierung der Wirkung zur konkreten Observable zur Bestimmung des statischen Potentials. Kapitel 3 enthält dann die Details der konkreten Implementation und den Ablauf der Simulation und in Kapitel 4 werden schließlich die gewonnenen Daten ausgewertet und die Ergebnisse

präsentiert.

Theorie

2.1 Vom Pfadintegral zum Metropolis-Algorithmus

2.1.1 Feynman

Ausgangspunkt für die verwendete Methode ist der im Rahmen des Feynman'schen Pfadintegralformalismus eingeführte Propagator [3]

$$\langle f | e^{-Ht} | i \rangle = \int \mathcal{D}[x] e^{-S[x]}, \quad (2.1)$$

welcher den Übergang eines physikalischen Systems charakterisiert durch den Hamiltonian H resp. der Wirkung S beschreibt. $x(t)$ sei hier eine Funktion, welche die Trajektorie des Systems vom Anfangszustand $|i\rangle$ in den Endzustand $|f\rangle$ charakterisiert, $\int \mathcal{D}[x]$ bezeichne die Integration über alle Trajektorien. Die berechnete Amplitude setzt sich also aus allen möglichen Pfaden zusammen, die das System wählen kann, jeder dieser Pfade $x(t)$ wird mit dem exponentiellen Faktor $e^{-S[x]}$ gewichtet.

In der hier notierten Form wurde $\hbar = 1$ angenommen sowie die Wick-Rotation zu imaginärer Zeit

$$t \mapsto it$$

verwendet. Letzteres hat den Vorteil, dass anstelle von (womöglich stark oszillierenden) Phasen nur exponentielle Faktoren betrachtet werden müssen, was insb. die Berechnungen mit numerischen Methoden vereinfacht. [3]

Der Erwartungswert einer Observablen O ist im Rahmen dieses Formalismus' wie folgt definiert [3]:

$$\langle O \rangle = \frac{1}{Z} \int \mathcal{D}[x] O[x] e^{-S[x]}.$$

Hierbei ist

$$Z = \int \mathcal{D}[x] e^{-S[x]}.$$

Solche Observablen O sollen schlussendlich gemessen werden. Es fällt auf, dass diese Definition der

eines Erwartungswertes von $O[x]$ bzgl. der Verteilung

$$P^{\text{Feyn}} \mathcal{D}[x] = \frac{1}{Z} e^{-S[x]} \mathcal{D}[x] \quad (2.2)$$

entspricht. Um diesen Erwartungswert anzunähern, sollen Trajektorien $\{x^\alpha\}$ aus dieser Verteilung gezogen werden. Dann kann $\langle O \rangle$ über das arithmetische Mittel zu

$$\bar{O} = \sum_{\alpha} O[x^\alpha]$$

abgeschätzt werden. $\langle O \rangle$ kann durch Erhöhung der Stichprobe beliebig gut durch \bar{O} approximiert werden.

Dieser Ansatz vereinfacht die gestellte Aufgabe enorm: Anstelle der Berechnung eines unendlichdimensionalen Integrals zum Finden von $\langle O \rangle$ müssen lediglich Trajektorien $\{x^\alpha\}$ aus der Verteilung gezogen werden. Das soll das nächste Etappenziel sein.

2.1.2 Markov¹

Eine Möglichkeit, Elemente aus einer bestimmten Verteilung zu generieren bietet ein iterativer stochastischer Prozess, der als Markov-Kette bezeichnet wird. Die Betrachtung hier ist an die von Freedman und Creutz [3] angelehnt.

Sei zunächst x eine kontinuierliche Größe, welche den Zustand eines stochastischen Systems beschreibt. $W(x, x')$ ist dann eine Funktion, welche die Übergangswahrscheinlichkeit des Systems vom Zustand x in den Zustand x' wiedergibt. Da es sich um eine Wahrscheinlichkeitsdichte handelt, muss

$$W(x, x') \geq 0 \quad \forall x, x' \quad \text{und} \quad \int dx' W(x, x') = 1 \quad \forall x \quad (2.3)$$

gelten. Die zweite Gleichung entspricht der Aussage, dass das System immer in irgendeinen Zustand übergehen muss.

Ein n -schrittiger Übergangsprozess von x nach x' über $n - 1$ Zwischenzustände lässt sich dann durch folgendes Integral beschreiben:

$$\begin{aligned} W^{(n)}(x, x') &= \int dx_1 dx_2 \dots dx_{n-1} W(x, x_1) W(x_1, x_2) \dots W(x_{n-1}, x') \\ &= \int d\tilde{x} W^{(n-1)}(x, \tilde{x}) W(\tilde{x}, x'). \end{aligned}$$

Nun lässt sich zeigen, dass für große n ein Gleichgewichtszustand erreicht wird, der nicht mehr vom Anfangszustand abhängt [3]:

$$\lim_{n \rightarrow \infty} W^{(n)}(x, x') = P(x').$$

Den Beweis hierfür liefern Freedman und Creutz [3]. $P(x')$ ist dann ein „Eigenvektor“ von $W(x, x')$,

1

wie sich durch iteratives Einsetzen der Definition leicht nachvollziehen lässt:

$$P(x') = \lim_{n \rightarrow \infty} P(x) W^{(n+1)}(x, x') = \lim_{n \rightarrow \infty} \int dx_n W^{(n)}(x, x_n) W(x_n, x') = \int d\tilde{x} P(\tilde{x}) W(\tilde{x}, x').$$

Daher auch die Bezeichnung als Gleichgewichtszustand: Haben wir ein Verfahren, dass x' aus x erzeugt, wobei die für $W(x, x')$ geforderten Eigenschaften erfüllt werden, und ist die Konfiguration x erst einmal in einem Zustand, der durch $P(x)$ beschrieben wird, gilt dies auch für alle durch weitere Iterationen erzeugten Zustände x' . Indes muss natürlich nicht $x' = x$ gelten, lediglich $P(x') = P(x)$ ist garantiert.

Da $W(x, x')$ eine Wahrscheinlichkeitsdichte ist, gilt dies auch für $P(x')$. Die Idee ist nun, $W(x, x')$ so zu gestalten, dass wir nach vielen Iterationen von $W(x, x')$ die gewünschte Verteilung P^{Feyn} (2.2) erhalten. Hierfür wollen wir zunächst die Eigenschaft der sog. *detailed balance* einführen. Sie besagt [3]:

$$\frac{W(x, x')}{W(x', x)} = \frac{P(x')}{P(x)} \quad (2.4)$$

Hierdurch wird die o. g. Eigenwertgleichung impliziert:

$$\int dx W(x, x') = \int dx P(x) \frac{P(x')}{P(x)} W(x', x) = P(x') \underbrace{\int dx W(x', x)}_{=1 \text{ (2.3)}}.$$

Setzen wir in die detailed balance Bedingung (2.4) die Definition der gewünschten Wahrscheinlichkeitsdichte (2.2) ein, so finden wir

$$\frac{W(x, x')}{W(x', x)} = \frac{e^{-S[x']}}{e^{-S[x]}} =: e^{-\Delta S[x', x]}$$

als Anforderung an das Verfahren zur Erzeugung von x' aus x .

2.1.3 Metropolis

So ein Verfahren ist der Metropolis-Algorithmus. Seine grundsätzliche Funktionsweise ist in Listing 1 nachzulesen. Um die Funktionsweise zu verstehen, ist es wichtig zu verstehen, dass in der numerischen Praxis eine Trajektorie x üblicherweise in irgendeiner Form als ein Array mit Elementen $x_i \in x$ gespeichert wird.

Alle Elemente von $x_i \in x$ werden also nacheinander zunächst zufällig verändert, dann wird jeweils berechnet, wie sich dadurch die Wirkung verändert. Wird sie kleiner, setzt man x_i auf den neuen Wert. Für den Fall $\Delta S > 0$, kann x_i trotzdem aktualisiert werden, dann allerdings mit der Wahrscheinlichkeit $P = e^{-\Delta S}$. (Dies wird in der Praxis durch die zufällige Ziehung von r aus $(0, 1)$ und den Vergleich mit P bewerkstelligt.) Zusammen gilt also für einen einzelnen Schritt:

$$W(x, x') = \begin{cases} 1 & \text{wenn } \Delta S < 0, \\ e^{-\Delta S} & \text{sonst.} \end{cases}$$

Hiermit wird auch klar, dass der Algorithmus die detailed balance-Bedingung (2.4) erfüllt: Sei oBdA

Algorithm 1 Metropolisalgorithmus

Require: x

```
for  $x_i \in x$  do
  generate random  $x'_i$ 
  compute  $\Delta S[x'_i, x_i, x] = S[\{\dots, x_{i-1}, x'_i, x_{i+1}, \dots\}] - S[\{\dots, x_{i-1}, x_i, x_{i+1}, \dots\}]$ 
  if  $\Delta S[x'_i, x_i, x] < 0$  then
    set  $x_i = x'_i$ 
  else
    draw random  $r \in U(0, 1)$ 
    if  $r < e^{-\Delta S[x'_i, x_i, x]}$  then
      set  $x_i = x'_i$ 
    end if
  end if
end for
```

 $S[x'] < S[x]$. Dann ist

$$W(x, x') = 1, \quad W(x', x) = e^{-(S[x] - S[x'])} \rightarrow \frac{W(x, x')}{W(x', x)} = \frac{e^{-S[x']}}{e^{-S[x]}}.$$

In der Praxis ist der Algorithmus noch in dreierlei Hinsicht verändert: Der wichtigste Aspekt ist der, dass die *Veränderung* der Wirkung ΔS durch die Modifikation eines einzelnen Gitterpunkts x_i oft nur von x_i, x'_i und den direkten „Nachbarn“ (je nach Topologie des Problems) von x_i abhängt. Dies vereinfacht die auszuführende Rechnung je Gitterpunkt enorm. Im Gegensatz zum notierten Algorithmus werden die neuen Vorschläge x'_i außerdem „nah“ an den ursprünglichen Positionen x_i gewählt (die Metrik hängt vom betrachteten Problem ab). Dadurch wird zwar die potentielle Veränderung der Trajektorie durch eine Iteration kleiner, allerdings erhöht sich so die Wahrscheinlichkeit, dass der neue Vorschlag x_i vom Algorithmus „angenommen“ wird. Schließlich werden für jedes x_i mehrere Iterationen des Algorithmus ausgeführt, bevor zum nächsten Gitterpunkt übergegangen wird. Hierdurch wird x_i zunächst mit seinen „Nachbarn“ optimal eingestellt, was auch die allgemeine Konvergenz zum Equilibrium beschleunigt.

2.2 Intermezzo: Der harmonische Oszillator als Beispiel

Um das zuvor Beschriebene etwas besser greifbar zu machen, soll es im Folgenden kurz anhand des quantenmechanischen harmonischen Oszillators erläutert werden. Die Wirkung des harmonischen Oszillators lautet nach Anwendung der Wick-Rotation²

$$S[x] = \int_0^T dt \left\{ \frac{m}{2} \dot{x}(t)^2 + \frac{\omega^2}{2} x(t)^2 \right\},$$

²Man bemerke den relativen Faktor +1 zwischen kinetischem und Potentialterm, anders als im Fall mit Minkowski-Raumzeit!

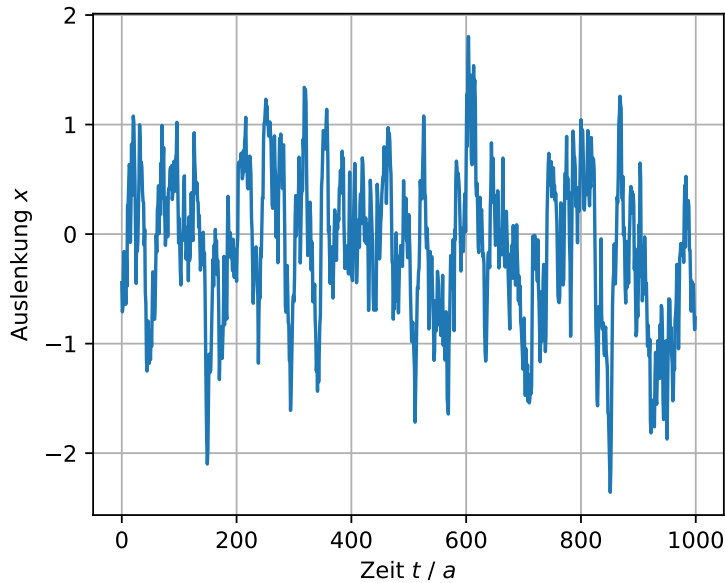


Abbildung 2.1: Ein Beispiel für eine Trajektorie des harmonischen Oszillators nach 10000 sweeps des Metropolisalgorithmus’.

Die Trajektorien x werden einfach als eindimensionales Array gespeichert, der Einfachheit halber nehmen wir $m = \omega = 1$ an. Dann lautet die auf dem Gitter formulierte, diskretisierte Version der Wirkung

$$S[x] = \frac{a}{2} \sum_{i=1}^N \left\{ \left(\frac{x_i - x_{i-1}}{a} \right)^2 + x_i^2 \right\}.$$

Es ist klar, dass für $\Delta S[x'_i, x_i, x]$ nur wenige Terme aus dieser Summe benötigt werden. In der Simulation wurden insgesamt 10000 Sweeps durchgeführt, wobei das System „kalt“, also mit $x_i = 0 \forall x_i \in x$ initialisiert wurde. Eine Trajektorie am Ende dieser 10000 sweeps ist exemplarisch in Abb. ?? dargestellt. Es wird erwartet, dass sich das System nach vielen Iterationen im Grundzustand befinden wird. Dies lässt sich durch folgende Überlegung motivieren: Füge in den Propagator (2.1) einen Einheitsoperator, ausgedrückt durch die Energieeigenzustände, $\mathbb{1} = \sum_n |n\rangle \langle n|$ ein. Es ergibt sich

$$\langle x_f | e^{-Ht} | x_i \rangle = \sum_n \langle x_f | e^{-Ht} | n \rangle \langle n | x_i \rangle = \sum_n e^{-E_n t} \langle x_f | n \rangle \langle n | x_i \rangle \xrightarrow{t \rightarrow \infty} e^{-E_0 t} \langle x_f | 0 \rangle \langle 0 | x_i \rangle. \quad (2.5)$$

Für große t , d. h. hier nach vielen Iterationen sollte das System im niedrigst liegenden Energieeigenzustands liegen. Das wollen wir überprüfen, indem wir die Energie sowie die Wellenfunktion des Zustands aus den Daten der Simulation extrahieren.

Die Energie ist hier die simplere Sache: Hierfür messen wir zwischen den Sweeps die Energie

$$\bar{E} = \hbar \omega \overline{x^2} = \overline{x^2},$$

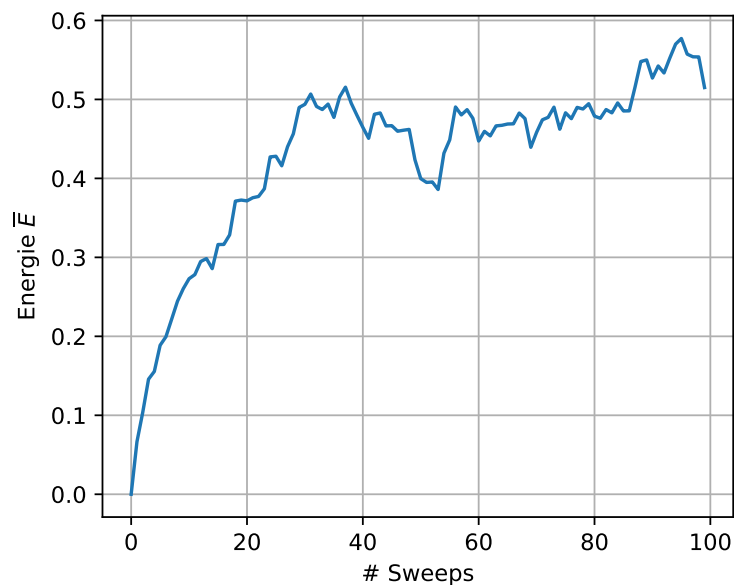


Abbildung 2.2: Die gemessenen Energien \bar{E} als Funktion der Monte-Carlo-Zeit. Es ist klar eine Konvergenz zu einem Plateau bei ≈ 0.5 erkennbar.

welche der Gesamtwirkung des Systems entspricht³. Das Ergebnis ist in Abb. ?? dargestellt: Zunächst ist deutlich sichtbar, wie das System vom „kalten“ Anfangszustand in den Gleichgewichtszustand konvergiert. Die Werte zum Berechnen der Grundzustandsenergie nehmen wir ab 1000 sweeps und messen nur alle 50 sweeps, um die Korrelation zwischen den Werten zu verringern. Wir finden

$$\bar{\bar{E}} = 0.491 \pm 0.017,$$

was den theoretischen Wert $E_0 = 0.5$ in den Fehlerbereich miteinschließt.

Die Grundzustandswellenfunktion ψ_0 lässt sich zwar nicht als eine direkte Observable messen, mittels eines Tricks, können wir ihr Betragsquadrat $|\psi_0|^2$ aber dennoch aus den Daten extrahieren. Hierfür nehmen wir einfach alle Positionen aus allen Sweeps und stellen sie in einem Histogramm dar. Da $|\psi_0|^2$ die Wahrscheinlichkeitsdichte beschreibt, sollte das Histogramm ebendiese Verteilung widerspiegeln. In Abb. ?? ist beides dargestellt, das erwartete Verhalten ist eingetreten.

³In dieser Rechnung wurde der Virialsatz miteinbezogen: $T = \frac{1}{2}xV'(x)$. [3]

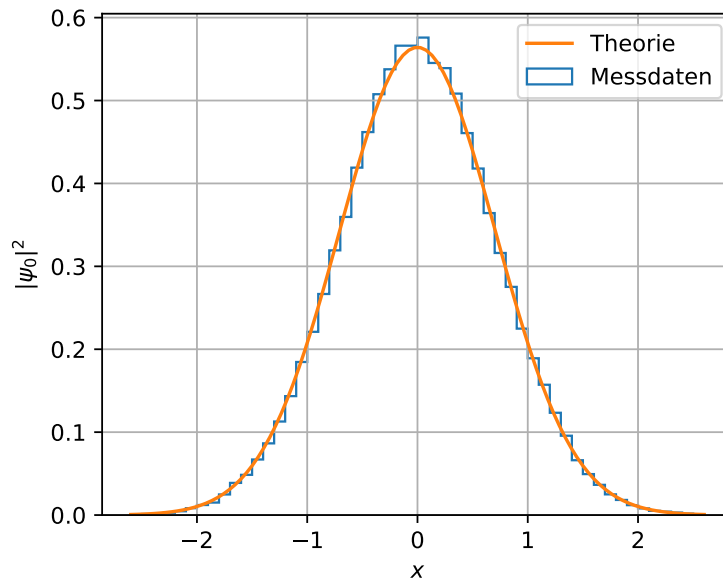


Abbildung 2.3: Zustandsdichte des Grundzustands des harmonischen Oszillators.

2.3 Das SU(2)-Eichfeld auf einem Gitter

2.3.1 Yang-Mills

Hauptbetrachtungsobjekt dieser Arbeit soll das Eichfeld A_μ der SU(2)-Eichsymmetrie sein. Der Lagrangian einer betrachteten Theorie enthält Terme

$$\bar{\psi}(x)(i\gamma^\mu \partial_\mu - m)\psi(x),$$

bei ψ handelt es sich also um ein fermionisches Feld⁴. Die betrachtete Eichtransformation wirkt auf die Felder dann wie folgt [4]:

$$\psi(x) \mapsto \Omega(x)\psi(x), \quad \Omega(x) \in \text{SU}(2).$$

Damit die Theorie eichinvariant wird, soll die Ableitung der Felder genauso transformieren wie die Felder selbst und dafür muss ∂_μ durch eine kovariante Ableitung $D_\mu = \partial_\mu + iA_\mu$ ersetzt werden. Setzen wir dann als Transformationsverhalten für das Eichfeld

$$A_\mu(x) \mapsto \Omega(x)A_\mu(x)\Omega(x)^\dagger + i \left(\partial_\mu \Omega(x) \right) \Omega(x)^\dagger$$

⁴Hinzuzufügen ist, dass es sich hierbei um eine stark vereinfachte Version handelt: in der Realität wäre noch die Farbladung zu berücksichtigen. [4]

an [4], so finden wir unter Verwendung von $\Omega^\dagger \Omega = 1$

$$\begin{aligned} D_\mu(x)\psi(x) &\mapsto \left(\partial_\mu + i\Omega(x)A_\mu\Omega(x)^\dagger - \left(\partial_\mu\Omega(x)\right)\Omega(x)^\dagger\right)\Omega(x)\psi(x) \\ &= \left(\partial_\mu\Omega(x)\right)\psi + \Omega(x)\partial_\mu\psi(x) + i\Omega(x)A_\mu(x)\psi(x) - \left(\partial_\mu\Omega(x)\right)\psi \\ &= \Omega(x)D_\mu(x)\psi(x). \end{aligned}$$

So kann also die Eichinvarianz der Theorie erreicht werden.

Die Dynamik des Eichfelds wird typischerweise durch die Yang-Mills-Wirkung

$$S[A] = \frac{1}{2g^2} \int d^4x \operatorname{tr} [F_{\mu\nu}(x)F^{\mu\nu}(x)] \quad (2.6)$$

beschrieben. [4] Hierbei ist

$$F_{\mu\nu}(x) = -i[D_\mu(x), D_\nu(x)] = \partial_\mu A_\nu(x) - \partial_\nu A_\mu(x) + i[A_\mu(x), A_\nu(x)]$$

der Feldstärketensor. Zunächst soll nun eine Version dieser Wirkung auf einem Gitter definiert werden.

2.3.2 Wilson

Um die Integration über alle Dimensionen der Raumzeit zu implementieren, brauchen wir ein vierdimensionales Gitter. Dieses Gitter habe in Zeitrichtung eine Ausdehnung von L_t Gitterpunkten und in die drei Raumrichtungen jeweils eine Ausdehnung von L_s Gitterpunkten. Wir verwenden periodische Randbedingungen: man kann das Gitter nicht „verlassen“, sondern läuft immer auf der gegenüberliegenden Seite wieder hinein. Befindet man sich z. B. bei $(L_t - 1, 0, 0, 0)$ und geht einen Schritt in Zeitrichtung, so befindet man sich danach wieder bei $(0, 0, 0, 0)$. Das Gitter ist gewissermaßen in jeder Dimension an den Enden zusammengeklebt:

$$\Lambda = \{0, 1, \dots, L_t - 1\} / (0 \sim L_t - 1) \times (\{0, 1, \dots, L_s - 1\} / (0 \sim L_s - 1))^3,$$

es handelt sich also um einen vierdimensionalen, diskretisierten Hypertorus. Der Abstand der Gitterpunkte sei a , als Basis für das Gitter wählen wir die Vektoren a_μ , die von einem Gitterpunkt auf den nächsten in Richtung μ zeigen. Sie haben folglich die Länge a und die Basis ist demnach keine Orthonormalbasis.

Nun wollen wir unsere Felder auf dem Gitter definieren. Die kanonische Entscheidung wäre, ihren Definitionsbereich auf das Gitter zu beschränken. Leider zeigt sich bereits beim fermionischen Feld, dass es so einfach nicht ist. Die diskretisierte Ableitung ließe sich z. B. so umsetzen:

$$\partial_\mu \psi(x) = \frac{\psi(x + a_\mu) - \psi(x)}{a}.$$

Betrachten wir dann das Verhalten so einer Ableitung unter der Eichtransformation, so finden wir

$$\partial_\mu(x)\psi(x) \mapsto \frac{\Omega(x + a_\mu)\psi(x + a_\mu) - \Omega(x)\psi(x)}{a}.$$

Dieses Ergebnis ist insofern problematisch, als dass wir durch die oben so geschickt eingeführte kovariante Ableitung D_μ nicht das gewünschte Transformationsverhalten erreichen können. Das liegt primär daran, dass hier Ω an zwei verschiedenen Orten, x und $x + a_\mu$, verwendet wird. Die Lösung findet sich in sog. Paralleltransporten $U_\mu(x) \in \text{SU}(2)$, deren Transformationsverhalten durch

$$U_\mu(x) \mapsto \Omega(x) U_\mu(x) \Omega(x + a_\mu)^\dagger$$

definiert ist. [4] Verwenden wir dann die folgende neue kovariante Ableitung

$$\tilde{D}_\mu \psi(x) = \frac{U_\mu(x) \psi(x + a_\mu) - \psi(x)}{a},$$

so erhalten wir wieder das gewünschte Transformationsverhalten

$$\begin{aligned} \tilde{D}_\mu \psi(x) &\mapsto \frac{\overbrace{\Omega(x) U_\mu(x) \Omega(x + a_\mu)^\dagger}^{=1} \Omega(x + a_\mu) \psi(x + a_\mu) - \Omega(x) \psi(x)}{a} \\ &= \Omega(x) \tilde{D}_\mu \psi(x). \end{aligned}$$

Die Paralleltransporte verdienen noch etwas Aufmerksamkeit: $U_\mu(x)$ kann als „Zeiger“ von x nach $x + a_\mu$ interpretiert werden. Umgekehrt zeigt $U_\mu(x)^\dagger$ von $x + a_\mu$ auf x . [5] Um die Verbindung zum Eichfeld $A_\mu(x)$ herzustellen, reicht ein Blick auf ihre genaue mathematische Definition [4]

$$U_\mu(x) = P \exp \left(i \int_\gamma ds A_\mu \right),$$

hier ist γ ein Pfad, der x mit $x + a_\mu$ verbindet. Wenn man die direkte Verbindungslinie für γ annimmt und das Integral nun **in zu erster Ordnung in a auf dem Gitter ausführt**, so findet man [4]

$$U_\mu(x) = \exp \left(i a A_\mu(x) \right). \quad (2.7)$$

Dieser Zusammenhang wird im Folgenden nützlich sein bei der Betrachtung der Wilson-Wirkung, welche das Verhalten des Eichfelds auf einem Gitter beschreibt [5]:

$$S_{\text{Wilson}} = \frac{\beta}{2} \sum_{x \in \Lambda} \sum_{\mu < \nu} \text{Re tr} \left[\mathbb{1} - U_{\mu\nu}(x) \right]. \quad (2.8)$$

Üblicherweise wird in dieser Formulierung die inverse Kopplung $\beta = \frac{2N_c}{g^2} = \frac{4}{g^2}$ verwendet. **Was ist hier $U_{\mu\nu}$? Es handelt sich** um den kleinstmöglichen geschlossenen Ring von Paralleltransporten [5]

$$U_{\mu\nu}(x) = U_\mu(x) U_\nu(x + a_\mu) U_\mu(x + a_\nu)^\dagger U_\nu(x)^\dagger,$$

der auch *Plaquette* genannt wird. Man sieht leicht, dass die Spur⁵ dieser Plaquette (wie die aller

⁵In dieser Arbeit sind alle betrachteten Observablen X die Spur einer Funktion von Paralleltransporten $X = \text{tr } Y$. Deswegen ist stets die Spur $\text{tr } Y$ gemeint, auch wenn nur von Y als Observable gesprochen wird. Die Spur ist immer

geschlossenen Ringe von Paralleltransporten) eichinvariant ist, was auch die formulierte Wilson-Wirkung eichinvariant macht. Nun wollen wir zeigen, dass die Wilson-Wirkung für kleine Gitterabstände a wieder in die Yang-Mills-Wirkung (2.6) übergeht.

Dazu betrachte zunächst die Plaquette im Rahmen des oben gefundenen exponentiellen Ausdrucks für die Paralleltransporte (2.7):

$$\begin{aligned}
 U_{\mu\nu}(x) &= \exp\left(iaA_\mu(x)\right) \exp\left(iaA_\nu(x+a_\mu)\right) \exp\left(-iaA_\mu(x+a_\nu)\right) \exp(-iaA_\nu(x)) \\
 &= \exp\left\{ia\left(A_\mu(x) + A_\nu(x+a_\mu) - A_\mu(x+a_\nu) - A_\nu(x)\right)\right. \\
 &\quad + \frac{ia^2}{2}\left(-[A_\mu(x), A_\nu(x+a_\mu)] + [A_\mu(x), A_\mu(x+a_\nu)]\right. \\
 &\quad + [A_\mu(x), A_\nu(x)] + [A_\nu(x+a_\mu), A_\mu(x+a_\nu)] \\
 &\quad \left. + [A_\nu(x+a_\mu), A_\nu(x)]\right) + O(a^3)\left\} \\
 &= \exp\left\{ia \cdot 0 + ia^2\left(\partial_\mu A_\nu(x) - \partial_\nu A_\mu(x) + i[A_\mu(x), A_\nu(x)]\right) + O(a^3)\right\} \\
 &= \exp\left(ia^2 F_{\mu\nu}(x) + O(a^3)\right).
 \end{aligned}$$

Hierbei wurde eine Version der Baker-Campbell-Hausdorff-Regel [4]

$$e^A e^B = e^{A+B+\frac{1}{2}[A,B]+\dots}$$

verwendet. Im zweiten Schritt wurden die Felder um x entwickelt, also z. B.

$$A_\mu(x+a_\nu) = A_\mu(x) + a\partial_\nu A_\mu(x) + O(a^2).$$

Setzen wir das gefundene Ergebnis in die Wilson-Wirkung (2.8) ein, so erhalten wir unter Berücksichtigung der Antisymmetrie des Feldstärketensors [4]

$$\begin{aligned}
 S_{\text{Wilson}} &= \frac{\beta}{2} \sum_{x \in \Lambda} \sum_{\mu < \nu} \text{tr} \left[\mathbb{1} - \mathbb{1} + \frac{1}{2}(a^2 F_{\mu\nu}(x))^2 + O(a^8) \right] \\
 &= \frac{a^4}{2g^2} \sum_{x \in \Lambda} \text{tr} \left[\sum_{\mu, \nu} (F_{\mu\nu}(x))^2 \right] + O(a^8).
 \end{aligned}$$

Die Wilson-Wirkung bietet also eine ordentliche Annäherung an die im Kontinuum formulierte Yang-Mills-Wirkung.

notwendig, um mit ihrer zyklischen Eigenschaft die Eichinvarianz der Observablen zu erreichen.

2.3.3 Haar

Für die Berechnung von Erwartungswerten auf dem Gitter wird ein Integrationsmaß für die Paralleltransporte benötigt. Die naheliegende Wahl wäre [4]

$$\mathcal{D}[\mathcal{U}] = \prod_{x \in \Lambda} \prod_{\mu=0}^3 dU_{\mu}(x).$$

Hier ist $\mathcal{U} = \{U_{\mu}(x) | x \in \Lambda, \mu \in \{0, 1, 2, 3\}\}$ die Menge aller Paralleltransporte auf dem Gitter. Hiermit finden wir also z. B.

$$Z = \int \mathcal{D}[\mathcal{U}] e^{-S_{\text{Wilson}}[\mathcal{U}]}.$$

Auch für diese Größe wollen wir die Eigenschaft der Eichinvarianz ansetzen können, damit die schlussendlich berechneten Erwartungswerte ebenfalls eichinvariant sind. Da die Wilson-Wirkung bereits eichinvariant ist, muss also folgendes gelten:

$$dU'_{\mu}(x) = d\left(\Omega(x)U_{\mu}(x)\Omega(x+a_{\mu})^{\dagger}\right) \stackrel{!}{=} dU_{\mu}(x).$$

Der perfekte Kandidat für dieses Problem ist das sog. Haar-Maß, welches für Elemente einer Gruppe $U, V \in G$ definiert ist und folgende Bedingungen erfüllen muss [4]:

$$d(UV) = d(VU) = dU, \quad \int dU = 1.$$

Für eine etwas detailliertere Diskussion dieses Themas sei auf das Buch von Gattringer und Lang [4] verwiesen, hier soll uns genügen, dass das Integrationsmaß die genannten Bedingungen erfüllt und so unsere Erwartungswerte eichinvariant macht.

2.3.4 Wilson und Metropolis

Um den Metropolisalgorithmus auf dem Gitter umsetzen zu können, fehlt uns noch eine effiziente Methode, um ΔS zu berechnen. Es ist einleuchtend, dass die durch die Veränderung eines einzelnen Paralleltransportes $U_{\mu}(x)$ modifizierten Plaquettes ebenjene sind, an denen $U_{\mu}(x)$ als „Kante“ beteiligt ist. Je Ebene (μ, ν) in der Raumzeit gibt es hiervon zwei – „oberhalb“ und „unterhalb“ von $U_{\mu}(x)$. Wir

finden also

$$\begin{aligned}
 \Delta S &= -\frac{\beta}{2} \sum_{\nu \neq \mu} \text{tr} [U'_{\mu\nu}(x) - U_{\mu\nu}(x) + U'_{\mu\nu}(x - a_\nu) - U_{\mu\nu}(x - a_\nu)] \\
 &= -\frac{\beta}{2} \sum_{\nu \neq \mu} \text{tr} [U'_{\mu\nu}(x) - U_{\mu\nu}(x) + (U'_{\mu\nu}(x - a_\nu) - U_{\mu\nu}(x - a_\nu))^\dagger] \\
 &= -\frac{\beta}{2} \sum_{\nu \neq \mu} \text{tr} [(U'_\mu(x) - U_\mu(x)) U_\nu(x + a_\mu) U_\mu(x + a_\nu)^\dagger U_\nu(x)^\dagger \\
 &\quad + U_\nu(x - a_\nu) (U'_\mu(x) - U_\mu(x)) U_\nu(x + a_\mu - a_\nu)^\dagger U_\mu(x - a_\nu)^\dagger] \\
 &= -\frac{\beta}{2} (U'_\mu(x) - U_\mu(x)) \sum_{\nu \neq \mu} \text{tr} [U_\nu(x + a_\mu) U_\mu(x + a_\nu)^\dagger U_\nu(x)^\dagger \\
 &\quad + U_\nu(x + a_\mu - a_\nu)^\dagger U_\mu(x - a_\nu)^\dagger U_\nu(x - a_\nu)] . \\
 &\quad \quad \quad \underbrace{\hspace{15em}}_{=: K_{\mu\nu}(x)}
 \end{aligned}$$

Hier wurde im ersten Schritt $\text{tr } U = \text{tr } U^\dagger \ \forall U \in \text{SU}(2)$ und im zweiten Schritt die Zyklicität der Spur verwendet. Die Größe $K_{\mu\nu}(x)$ wird *staple* genannt. Wie zuvor angedeutet, muss also für die Berechnung von $\Delta S[U'_\mu(x), U_\mu(x), \mathcal{U}]$ nicht das gesamte Gitter berücksichtigt werden, es genügt die Betrachtung der $3 \cdot 6 = 18$ umliegenden Paralleltransporten.

2.3.5 Das statische Potential und Wilson-Loops

Eine wichtige Observable auf dem Gitter sind sog. Wilson-Loops, die geschlossene Ringe von Paralleltransporten darstellen. Im einfachsten Fall kann man sie sich als Rechteck vorstellen:

$$W(x, \mu, \nu, m, n) = \text{tr} \left[L(x, \mu, m) \cdot L(x + ma_\mu, \nu, n) \cdot L(x + na_\nu, \mu, m)^\dagger \cdot L(x, \nu, n)^\dagger \right],$$

$$L(x, \mu, m) = \prod_{i=0}^{m-1} U_\mu(x + ia_\mu)$$

Man erkennt gut, dass es sich hierbei um die Verallgemeinerung der zuvor besprochenen Plaquette $U_{\mu\nu}(x)$ handelt. Im Folgenden soll motiviert werden, dass sich aus dem Erwartungswert $\langle W(0, i, l, l_s) \rangle =: \langle W(r, t) \rangle$ das Potential zweier unendlich schwerer, immobiler („statischer“) Quarks mit Abstand $r = l_s a$ extrahieren lässt.

Ausgangspunkt ist die Überlegung (2.5), welche wir beim harmonischen Oszillator angestellt haben: für große Zeiten t wird sich ein physikalisches System im Rahmen des Feynman'schen Pfadintegralformalismus stets auf den mit den Grundzustand überlappenden Zustand übergehen. Nun ist also das Ziel, einen Quark-Antiquark-Paar-Zustand zu erzeugen, der bei konstantem Ort durch die Zeit propagiert und dann wieder annihiliert wird.

Die naheliegende Entscheidung liegt in einer Kombination des Propagators eines unendlich schweren Quarks an einer konstanten Position x [6]

$$G_\infty(t, \vec{x}) = U_0(t - a, \vec{x})^\dagger U_0(t - 2a, \vec{x})^\dagger \dots U_0(0, \vec{x})^\dagger$$

mit dem eines unendlich schweren Antiquarks, $G_\infty(t, \vec{y})^\dagger$. Hier haben wir bereits zwei „Kanten“ des späteren Wilson-Loops. Um das Konstrukt eichinvariant zu machen, fügen wir zwei weitere Kanten hinzu, die $(0, \vec{x})$ und $(0, \vec{y})$ bzw. (t, \vec{x}) und (t, \vec{y}) verbinden.⁶ So haben wir die zuvor definierten Wilson-Loops wiedergefunden und finden für ihren Erwartungswert [7]

$$\langle W(r, t) \rangle = C \cdot e^{-tV(r)}. \quad (2.9)$$

Im weiteren Verlauf dieser Arbeit werden wir also Wilson-Loops versch. Maße $\overline{W(r, t)}$ messen und durch Anpassungen von exponentiellen Modellen an die gewonnen Daten das statische Potential extrahieren.

Die hier vorgenommene Diskussion dieser Observablen erhebt keinerlei Anspruch auf Vollständigkeit oder Rigorosität. Die in dieser Hinsicht beste Betrachtung ist

⁶Hier sei hinzugefügt, dass diese beiden Kanten sich jeder Zeit durch sog. *gauge fixing* wieder „entfernen“ lassen, vgl. hierfür z. B. [4].

Methodik und Implementation

3.1 SU(2)-Matrizen

Nachdem die theoretischen Grundlagen gelegt wurden, kann mit der Implementation des Algorithmus begonnen werden. Zunächst wird hierfür eine numerische Darstellung von SU(2)-Matrizen benötigt. Die hier verwendete Form ist an die von Urbach und Petschlies [5] angelehnt. Aus der Definition von unitären Matrizen

$$U^\dagger U = U U^\dagger = \mathbb{1}$$

lässt sich ableiten, dass sich jede unitäre 2×2 -Matrix in der Form

$$U = \begin{pmatrix} a & b \\ -b^* & a^* \end{pmatrix}, \quad a, b \in \mathbb{C}, \quad |a|^2 + |b|^2 = 1.$$

darstellen lässt. [5]

Die Klasse speichert also lediglich die Einträge der oberen Zeile einer jeden Matrix. Man kann zeigen, dass sich Matrixmultiplikation im Rahmen dieser Darstellung umsetzen lässt [5]:

$$(a_1, b_1) \cdot (a_2, b_2) = (a_1 b_2 - b_1 b_2^*, a_1 b_2 + b_1 a_2^*).$$

Die weiteren Matrixoperationen ergeben sich trivial aus der verwendeten Form:

$$\begin{aligned} (a_1, b_1) + (a_2, b_2) &= (a_1 + a_2, b_1 + b_2), \\ (a, b)^\dagger &= (a^*, -b), \\ \text{tr}(a, b) &= 2 \operatorname{Re} a. \end{aligned}$$

Aufgrund der endlichen Präzision der verwendeten Datentypen muss für eine betrachtete Matrix U hin und wieder $\det U = 1$ forciert werden. Die geschieht durch [5]

$$(a, b) \mapsto \left(\frac{a}{\sqrt{|a|^2 + |b|^2}}, \frac{b}{\sqrt{|a|^2 + |b|^2}} \right).$$

3.2 Zufällige SU(2)-Matrizen

Eine Herausforderung bildet die Generierung zufälliger Paralleltransporte $U'_\mu(x)$, die, wie in Abschnitt 2.1.3 erwähnt, „nah“ an $U_\mu(x)$ liegen sollen. Hierzu nutzen wir, dass SU(2) isomorph zur Sphäre S^3 ist vermöge des Isomorphismus [5]

$$\begin{aligned}\phi : S^3 &\rightarrow \text{SU}(2), \\ (x^0, \vec{x}) &\mapsto x^0 \mathbb{1} + i \vec{x} \vec{\sigma}.\end{aligned}$$

Für die Punkte auf S^3 verwenden wir Kugelkoordinaten

$$\begin{pmatrix} x^0 \\ x^1 \\ x^2 \\ x^3 \end{pmatrix} = \begin{pmatrix} \cos(\chi) \sin(\vartheta) \cos(\varphi) \\ \sin(\chi) \sin(\vartheta) \cos(\varphi) \\ \sin(\chi) \sin(\vartheta) \sin(\varphi) \\ \sin(\chi) \cos(\vartheta) \end{pmatrix}.$$

Folgende Zufallszahlen werden generiert:

$$\begin{aligned}\chi &\in (0, 2\pi \cdot \delta), \\ \varphi &\in (0, 2\pi), \\ \cos(\vartheta) &\in (-1, 1).\end{aligned}$$

Hier kann $\delta > 0$ beliebig klein gewählt werden, sodass die resultierende Matrix

$$\begin{aligned}R &= (x^0 + ix^3, x^2 + ix^1) \\ &= (\cos(\chi) + i \sin(\chi) \cos(\vartheta), \sin(\chi) \sin(\vartheta) \sin(\varphi) + i \sin(\chi) \sin(\vartheta) \cos(\varphi))\end{aligned}$$

nah an $\mathbb{1}$ liegt¹. Dann kann für ein gegebenes $U_\mu(x)$ der neue Kandidat durch

$$U'_\mu(x) = R \cdot U_\mu(x)$$

erzeugt werden.

3.3 Implementierung des Algorithmus' sowie der Observablen

Der beschriebene Algorithmus wurde in C++ implementiert, der Programmcode ist auf Github² einsehbar. Hier soll nur eine kurze Auflistung der wichtigsten Klassen und Funktionen folgen, die Details lassen sich in den Header files nachlesen.

Die Klasse `SU2matrix` ist die Repräsentation unitärer 2×2 -Matrizen, wie sie in Abschnitt 3.1 beschrieben ist. Es wird jeweils die obere Reihe der Matrix mit zwei Einträgen vom Typ

¹Es sei erwähnt, dass selbst bei $\delta = 1$ die generierten Punkte nicht isotrop auf S^3 verteilt sein werden. Das hängt damit zusammen, dass bei der Generierung in Kugelkoordinaten die Jacobi-Determinante berücksichtigt werden muss. (Eine isotrope Verteilung der einzelnen Winkel auf ihren Intervallen garantiert a priori keine isotrope Verteilung auf der Sphäre.) Für die Komponenten x^1, x^2 und x^3 ist dies so geschehen, daher die Generierung von Zahlen $\cos(\vartheta) \in (-1, 1)$.

²<https://github.com/s6hevonc/bachelorarbeit>

`std::complex<double>` gespeichert. Die Klasse enthält eine Methode `renormalise()`, welche die Forcierung der Determinante auf 1 umsetzt sowie eine Methode `dagger()` für die hermitesche Adjunktion. Für die Addition und Multiplikation wurden die entsprechenden Operatoren *überladen*, d. h. für zwei Instanzen A und B der Klasse kann ihre Summe resp. ihr Produkt durch `A + B` resp. `A * B` berechnet werden. Schließlich gibt es eine Funktion `randomSU2`, die eine zufällige SU(2)-Matrix zurückgibt; sie nimmt δ als Argument entgegen.

Die zweite Klasse heißt `Gaugeconfig`. Eine Instanz U beschreibt den Zustand des Eichfeldes in Form der Paralleltransporter auf einem geg. Gitter, im Wesentlichen handelt es sich also um ein fünfdimensionales Array der Maße $L_t \times L_s \times L_s \times L_s \times 4$. Die 4 Am Ende rührt daher, dass je Gitterpunkt x vier Paralleltransporter $U_\mu(x)$ vorliegen, einer in jede Raumzeitrichtung. Wichtigste Methode der Klasse ist die Überladung des `()`-Operators. Konkret kann auf $U_\mu(x)$ mit `U(x, mu)` zugegriffen werden. Hier sind auch direkt die periodischen Randbedingungen implementiert: `x` ist ein `std::vector<long int>`, der auch negative Einträge haben kann. Zugehörig zu `Gaugeconfig` ist noch die Funktion `hotStart`, welche zu gegebenen Gittermaßen L_t und L_s und dem Parameter δ eine Konfiguration U zurückgibt, welche an jeden Gitterpunkt zufällig generierte SU(2)-Matrizen enthält.

Die für den Metropolisalgorithmus notwendigen staples sind mittels der Funktion `getStaple` zu berechnen. Der fertige Algorithmus ist dann in der Funktion `sweep` implementiert, welche eine einzelne Iterationen auf einer gegebenen `Gaugeconfig` durchführt. Hierbei kann δ für die Zufallsgenerierung sowie die Anzahl der Iterationen je Paralleltransport spezifiziert werden.

Für die Observablen existiert zunächst die Funktion `getPlaquette`, welche die Plaquette ausgehend von einem geg. Gitterpunkt x zurückgibt. Die Funktion `gaugeEnergy` berechnet die gesamte Wirkung des Gitters. Weiterhin gibt die Funktion `getPlanarLoop` den Wert eines planaren Wilson-Loops zurück. Schließlich finden sich in `getSqrt2Loop`, `getSqrt3Loop` etc. einzelne Funktionen, die nichtplanare Wilson-Loops vermessen. So ist es möglich, auch nicht ganzzahlige r für den Abstand des Quark-Antiquark-Paares zu betrachten.

3.4 Parameter und Ablauf der durchgeführten Simulationen

Um einen Überblick über den Ablauf der Simulation zu bieten, ist in Listing 3.1 eine exemplarische `main`-Funktion nachzulesen.

Als erstes werden alle wichtigen Variablen deklariert und initialisiert. Bei `std::mt19937` handelt es sich um die C++-Standardimplementation eines Mersenne-Twister-Generators für Pseudozufallszahlen. Die Messungen werden auf einem Gitter der Maße $(L_t, L_s) = (10, 10)$ durchgeführt, welches zunächst kalt, also mit $\delta = 0 \rightarrow U_\mu(x) = 1 \forall x \forall \mu$ initialisiert wird. Als nächstes werden die Parameter für die einzelnen Sweeps ($\beta = 2.3, \delta = 0.1$) sowie die Zahl der Hits je Paralleltransport sowie die Gesamtzahl der Iterationen festgelegt. Außerdem wird die Speicherung der Messwerte vorbereitet.

Nachdem das Gitter zunächst equilibriert wird, kann die Messphase beginnen. Der eigentliche Messvorgang wird, um das Verfahren zu beschleunigen, mit `forkomp` auf mehreren Kernen durchgeführt. Konkret wird die Schleife mit den Messungen mit dem Befehl in Zeile 37 parallelisiert. `reduction(merge: results)` bedeutet, dass die Ergebnisse aus allen threads am Ende im Array `results` zusammen gespeichert werden sollen. Außerdem ist es wichtig, dass alle threads auf die engine, die Instanz des Zufallsgenerators, gleichzeitig zugreifen können: Standardmäßig würde jeder thread seine eigene Instanz erhalten, dann wären aber alle in den einzelnen threads generierten Pseudozufallszahlen gleich und die resultierenden Daten korreliert. `num_threads` spezifiziert die

Listing 3.1: Ein Beispiel für main.cpp

```

1 int main()
2 {
3     // for random number generation:
4     std::random_device rd {}; // to generate the seed for...
5     std::mt19937 engine { rd() }; // Mersenne twister generator
6
7     // lattice parameters
8     const std::size_t timeSize { 10 };
9     const std::size_t spaceSize { 10 };
10
11    // initial configuration:
12    Gaugeconfig U { hotStart(timeSize, spaceSize, engine, 0.) };
13
14    // sweep parameters:
15    const double beta { 2.3 };
16    const double delta { .1 };
17    const std::size_t numberOfSweeps { 10 };
18    const std::size_t iterationsPerSight { 10 };
19
20    // to save observables:
21    const std::string dataDir { "../data/" };
22    const std::string filename { "wilsonsqrt.txt" };
23    std::vector<double> results;
24
25    std::cout << "warming up..." << std::endl;
26    for (std::size_t i {0}; i < 10; i++)
27    {
28        sweep(U, beta, delta, iterationsPerSight, engine);
29    }
30
31    std::cout << "performing " << numberOfSweeps << " sweeps..." << std::endl;
32
33    // define merge behaviour for std::vector:
34    #pragma omp declare reduction (merge : std::vector<double> : omp_out.insert(omp_out.end(), omp_in.
    ↪ begin(), omp_in.end()))
35    // parallelised loop: the results are 2 b merged, the engine has to be
36    // shared over all threads to obtain uncorrelated data, use two threads
37    #pragma omp parallel for reduction(merge: results), shared(engine), num_threads(2)
38        for (std::size_t i=0; i < numberOfSweeps; i++)
39        {
40            results.push_back(getSqrt2Loop(U, 2));
41
42            for (size_t j = 0; j < 5; j++)
43            {
44                sweep(U, beta, delta, iterationsPerSight, engine);
45            }
46        }
47    std::cout << "zero if writing successful: ";
48    std::cout << write_2d(results, dataDir + filename, ',') << '\n';
49    return 0;
50 }

```


Anzahl der threads. Zwischen jeder Messungen werden je fünf sweeps durchgeführt, um Korrelation zwischen den Daten zu verringern.

Nach Ablauf der Messphase werden die Daten schließlich noch gespeichert, um sie danach auswerten zu können. Für alle im folgenden Kapitel diskutierten Daten sind die Messwerte sowie die Parameter der dazugehörigen Simulationen im Anhang [B](#) zu finden.

Auswertung

4.1 Energie und durchschnittliche Plaquette

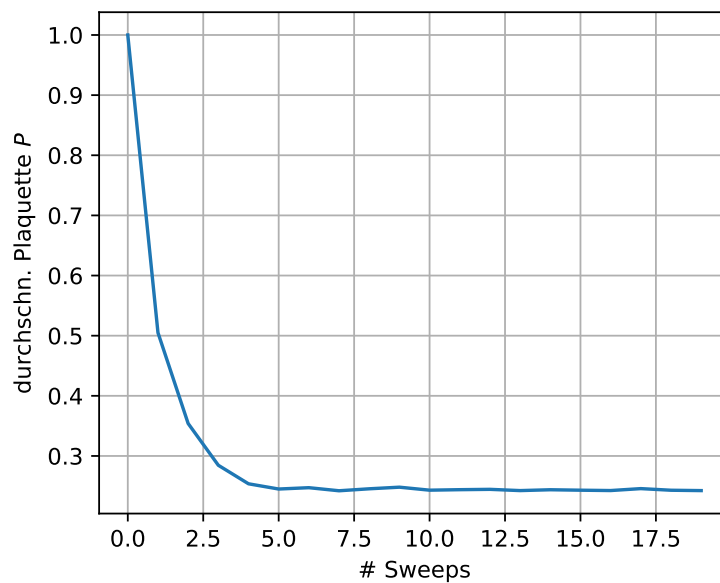


Abbildung 4.1: Die durchschnittliche Plaquette eines 16×8^3 -Gitters als Funktion der Iterationszahl: nach ca. 5 sweeps scheint das System bereits einen Gleichgewichtszustand zu erreichen.

Zunächst wurde die Gesamtenergie des Gitters

$$E = \sum_{x \in \Lambda} \sum_{\mu < \nu} \text{Re tr} U_{\mu\nu}(x)$$

vermessen, woraus sich auch die durchschnittliche Plaquette

$$P = \frac{E}{6L_t L_s^3 N_c}$$

berechnen lässt. Das Ergebnis ist in Abhängigkeit von der Sweepzahl in Abb. ?? dargestellt. Wie man erkennen kann, wurde das System kalt, d. h. mit $U_\mu(x) \forall x \in \Lambda \forall \mu$ initialisiert, daher beginnen die Werte bei $P = 1$. In unter 20 Sweeps konvergiert das System dann zu einem stabilen Plateau. Hier kann nun gemessen werden. Für $\beta = 1$ erhalten wir

$$\bar{P} = 0.24308 \pm 0.00010.$$

Dieser Wert allein ermöglicht zwar den Vergleich mit vorherigen Simulationen, bietet allein aber wenig Einblicke in die Physik. Dazu bieten erst die Wilson-Loops mehr Gelgenheit.

4.2 Vermessen von Wilson-Loops verschiedener Größe

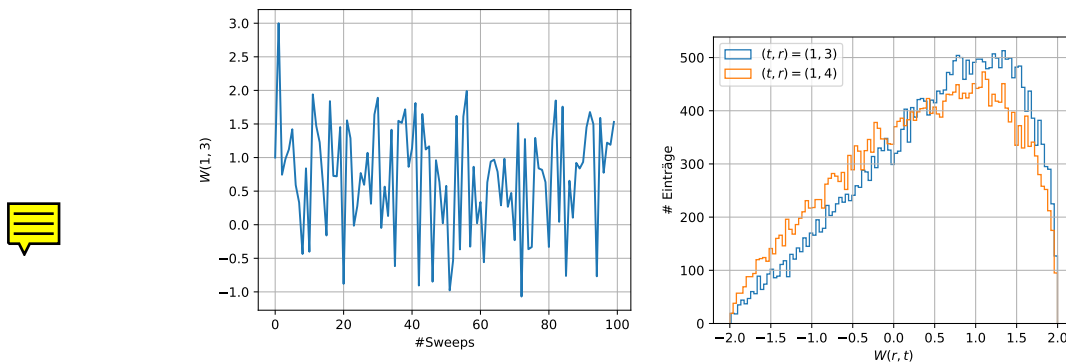


Abbildung 4.2: Links: Alle Messwerte eines 1×3 -Wilson-Loops. Rechts: Verteilung relevanter Messwerte (nach Equilibrierung alle 5 sweeps gemessen).

Dies war der nächste Schritt. Zunächst wurden Loops wie in Abschn. 2.3.5 definiert vermessen. Als Bsp. sind die Messwerte für einen 1×3 -Loop als Funktion der Iterationszahl in Abb. ?? dargestellt. Als erstes fällt auf, dass die Messwerte sehr stark zwischen den Werten ± 2 fluktuieren. Deswegen sind viele Messwerte zur zuverlässigen Bestimmung des Erwartungswertes notwendig. In ?? ist außerdem die Verteilung der Messwerte für zweimal 29800 Messungen dargestellt. Wieder ist zu erkennen, dass die Verteilung sehr „breit“ ist, und das insb. für die Unterscheidung der verschiedenen Messungen viele Messungen notwendig sind, um den Standardfehler auf $\overline{W(r,t)}$ entsprechend zu drücken. In diesem konkreten Fall wurde

$$\overline{W(1,3)} = 0.4984 \pm 0.0030, \quad \overline{W(1,4)} = 0.3270 \pm 0.0032$$

gemessen. Bemerkenswert ist, dass der Standardfehler in diesem Bsp. und bei allen Messungen (unabhängig von der Größe der Schätzer) ungefähr gleich groß ist. Dies ist vermutlich auf die Ähnlichkeit der Verteilung (und somit der Standardabweichung) und der gleichen Anzahl von Samples

zurückzuführen. (Der Standardfehler skaliert mit $\frac{\sigma}{\sqrt{N}}$.)

4.3 Extrahierung der Werte für das statische Potential

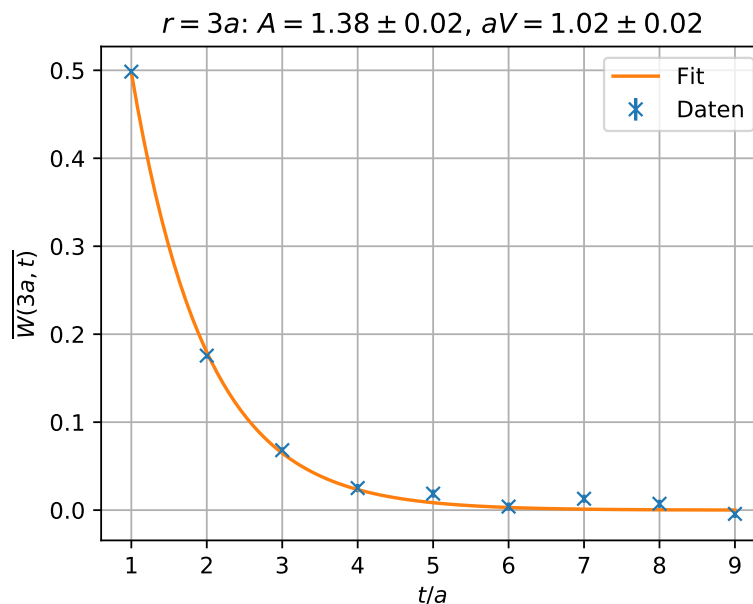


Abbildung 4.3: Beispiel für Wilson-Loop-Messwerte zur Bestimmung des statischen Potentials für $r = 3a$. Verwendetes Modell: $\overline{W(3a, t)} = A \cdot \exp(-aVt/a)$.

Grundlage für die Berechnung des Wertes des statischen **Potential** für ein festes r bietet die Relation (2.9). Zunächst wurde also der Erwartungswert $\overline{W(r, t)}$ in Abhängigkeit von t betrachtet, wie in Abb. ?? exemplarisch für $r = 3$ dargestellt ist. Der exponentielle Zusammenhang ist klar erkennbar, ein Fit liefert den gewünschten Parameter $aV(r)$. (t ist natürlich nur bis auf die Gitterkonstante a bestimmt.) Es fällt aber auch auf, dass die Messwerte bei größeren t nicht mehr allzu klar dem exponentiellen Zusammenhang folgen. Natürlich macht sich die Tatsache, dass der statistische Fehler immer ungefähr gleich groß ist, besonders bei kleinen Messwerten bemerkbar.

4.4 Confinement (?)

Die oben beschriebene Messung von $\overline{W(r, t)}$ wurde für $(r, t) \in \{1, \sqrt{2} \approx 1.4, \sqrt{3} \approx 1.7, 2, 3, 4\} \times \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ durchgeführt. Für die verschiedenen r -Werte wurde nach dem beschriebenen Verfahren $aV(r)$ bestimmt. Das Resultat ist in Abb. ?? dargestellt. Die Schlussfolgerung ist klar: Das Potential enthält deutlich einen linearen Term. Passt man eine Funktion der Form

$$aV(r) = -\frac{A}{r} + B + \sigma \cdot r$$



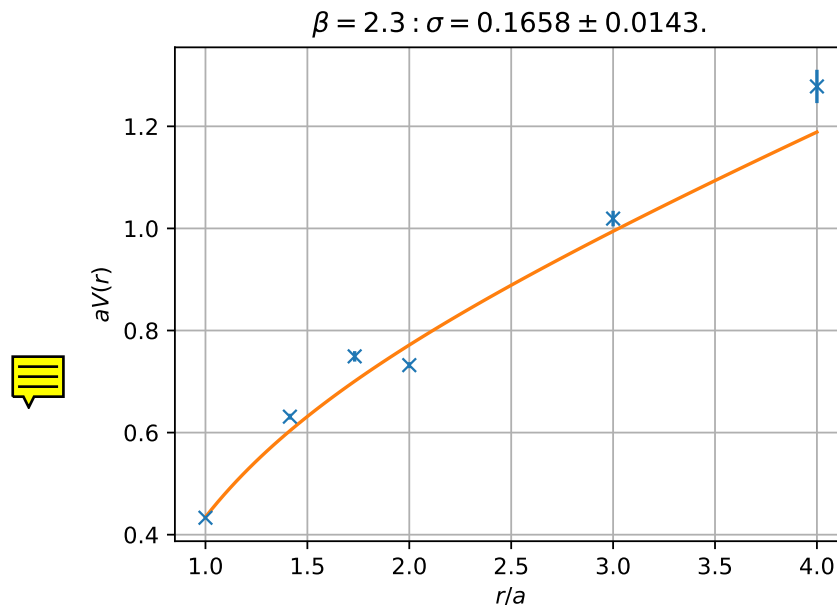


Abbildung 4.4: Das statische Potential in Abhängigkeit vom Abstand von Quark und Antiquark; es ist klar ein linearer Zusammenhang erkennbar.

an die Daten ab, so erhält man

$$\sigma = (0.166 \pm 0.014) \frac{1}{a}$$

für die sog. string tension¹. Natürlich ist dieser Graph bei Weitem nicht perfekt, der Fit könnte ebenfalls besser sein. Trotzdem kann ein linearer Zusammenhang erkannt werden.

Nun stellt sich natürlich die Frage nach den physikalischen Implikationen dieses Ergebnisses. Die SU(2)-Symmetrie ist nicht alleinige Wurzel einer einzelnen Wechselwirkung, vielmehr wird sie im Rahmen der Vereinigung der elektromagnetischen mit der schwachen Wechselwirkung verwendet, die von SU(2)×U(1) generiert wird. Diese Vereinheitlichung ist jedoch erst oberhalb einer bestimmten Energieskala resp. unterhalb einer bestimmten Längenskala sinnvoll. Gleichzeitig ist zu beachten, dass Confinement aufgrund der laufenden Kopplung nur bei kleinen Energien, i. e. großen Kopplungen auftritt. (Dies gilt auch für SU(3) und die starke Wechselwirkung: hier spricht man im Gegensatz zu Confinement bei kleinen Energien von *asymptotischer Freiheit* bei großen Energien, wo die Kopplung schwächer wird und Quarks nur noch leichtinteragierend sind.) Diese beiden Erkenntnisse zusammengekommen, ist zu schlussfolgern, dass bei den für die Realität relevanten Energieskalen Confinement nicht auftritt.

¹Das Potential hat Dimension L^{-1} , dementsprechend ist auch $[\sigma] = L^{-1}$.

Fazit

Um den Ansatz für die vorgelegte Demonstration von Confinement im Rahmen der $SU(2)$ -Eichsymmetrie nachvollziehen zu können, waren zunächst die theoretischen Hintergründe notwendig. Hierbei wurde als Erstes der Metropolis-Algorithmus vorgestellt, welcher das fundamentale Werkzeug zur numerischen Approximation beliebiger Feynman-Pfadintegrale darstellt. Damit wurde als nächstes exemplarisch der harmonische Oszillator vorgestellt, hier konnten Grundzustandsenergie und -wellenfunktion erfolgreich mit Simulationen angenähert werden.

Nun folgte die Betrachtung des von Wilson vorgelegten gitterfeldtheoretischen Pendantes zur Yang-Mills-Wirkung, welches die Dynamik des Eichfeldes der $SU(2)$ -Eichsymmetrie auf einem Gitter beschreibt. Dies ermöglichte dann die Betrachtung dieser Dynamik mit Hilfe des Metropolisalgorithmus sowie der Definition der Wilson-Loops als Observable zur Bestimmung des statischen Potentials eines Quark-Antiquark-Paares.

Die Beschränkung auf $SU(2)$ hatte den Vorteil, dass sich die Eichtransformationen und die gitterfeldtheoretische Formulierung des Eichfeldes in Form von Paralleltransporten relativ leicht als unitäre 2×2 -Matrizen mit Determinante 1 numerisch darstellen ließen. Die Implementierung erfolgte dann mit C++, es wurden Simulationen auf einem Gitter mit 10^4 Gitterpunkten zur Vermessung von Wilson-Loops verschiedener Maße vorgenommen.

Schließlich konnte im Rahmen der Auswertung der gemessenen Werte ebenjener lineare Term im statischen Potential wiedergefunden werden, den Wilson wie eingangs erwähnt ursprünglich vorhergesagt hatte. So konnte numerisch das Auftreten von Confinement im Rahmen der $SU(2)$ -Eichsymmetrie nachvollzogen werden.

Leider ist dieses Erkenntnis für die physikalische Realität nur bedingt insofern bedingt relevant, als dass die Energieskala, bei der Confinement auftritt weit unter der liegt, ab der die elektroschwache Vereinheitlichung ihren Gültigkeitsbereich hat, welche von $SU(2) \times U(1)$ generiert wird. Als nächster Schritt wäre deshalb eine Betrachtung der $SU(3)$ -Eichsymmetrie naheliegend, um tatsächlich Confinement für die starke Wechselwirkung zu demonstrieren.

Zusätzliche Graphen

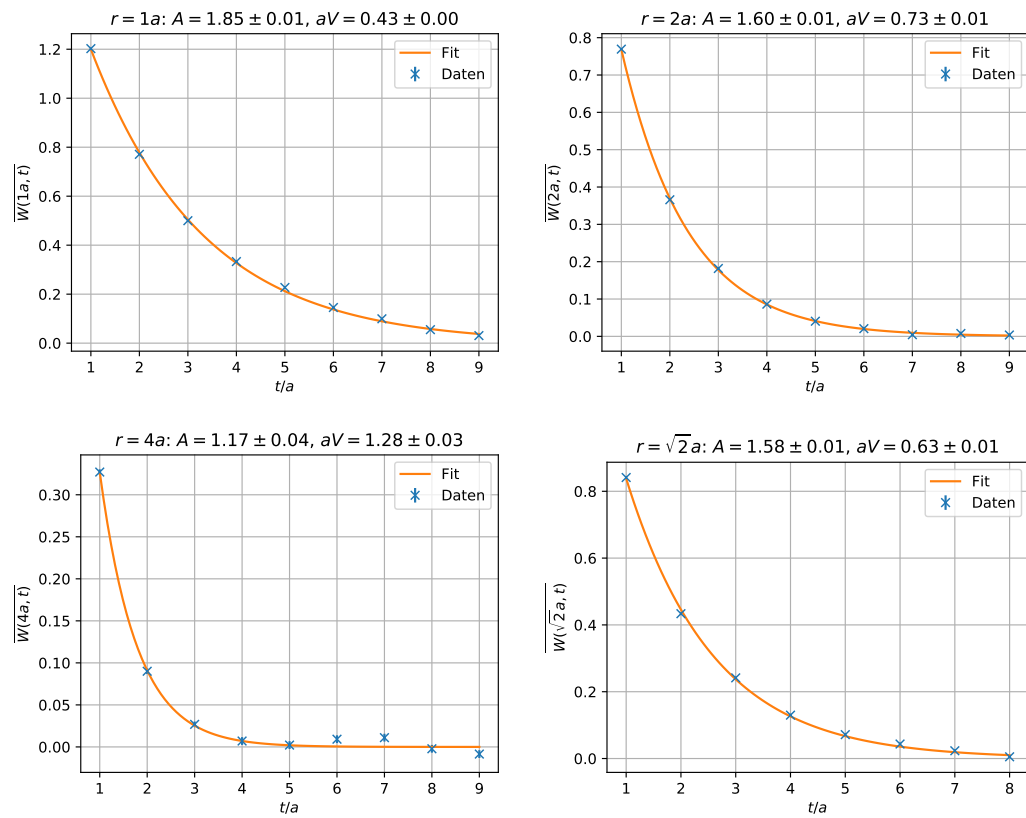


Abbildung A.1: Messwerte der Wilson-Loops bei $\beta = 2.3$ und festgehaltenem r . Verwendetes Modell: $\overline{W(r, t)} = A \cdot \exp(-aV(r)t/a)$.

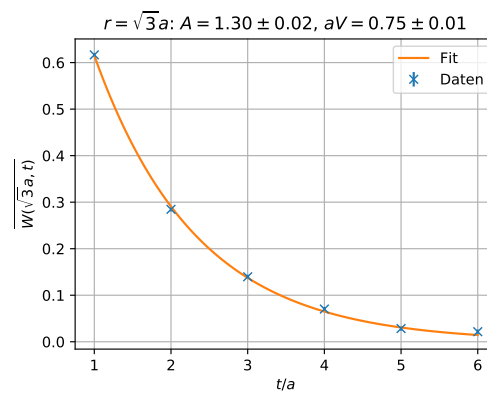


Abbildung A.2: Messwerte der Wilson-Loops bei $\beta = 2.3$ und festgehaltenem r . Verwendetes Modell: $\overline{W(r, t)} = A \cdot \exp(-aV(r)t/a)$.

Wertetabellen

t/a	$\overline{W(a,t)}$			t/a	$\overline{W(2a,t)}$			t/a	$\overline{W(3a,t)}$		
1	1.202	\pm	0.002	1	0.769	\pm	0.003	1	0.498	\pm	0.003
2	0.771	\pm	0.003	2	0.366	\pm	0.003	2	0.176	\pm	0.003
3	0.500	\pm	0.003	3	0.182	\pm	0.003	3	0.068	\pm	0.003
4	0.333	\pm	0.004	4	0.086	\pm	0.003	4	0.025	\pm	0.005
5	0.227	\pm	0.005	5	0.041	\pm	0.003	5	0.019	\pm	0.005
6	0.145	\pm	0.005	6	0.021	\pm	0.005	6	0.004	\pm	0.005
7	0.099	\pm	0.005	7	0.004	\pm	0.005	7	0.013	\pm	0.005
8	0.055	\pm	0.005	8	0.008	\pm	0.005	8	0.007	\pm	0.005
9	0.031	\pm	0.005	9	0.004	\pm	0.005	9	-0.004	\pm	0.005

Tabelle B.1: Wertetabellen für die Messungen der Wilson-Loops bei $\beta = 2.3$. Nach 1000 Sweeps zur Equilibrierung wurden die Messungen jeweils mit 5 sweeps Abstand genommen, um Korrelation zu reduzieren.

t/a	$\overline{W(4a,t)}$			t/a	$\overline{W(\sqrt{2}a,t)}$			t/a	$\overline{W(\sqrt{3}a,t)}$		
1	0.327	\pm	0.003	1	0.841	\pm	0.004	1	0.617	\pm	0.004
2	0.090	\pm	0.003	2	0.434	\pm	0.004	2	0.285	\pm	0.004
3	0.027	\pm	0.003	3	0.241	\pm	0.004	3	0.140	\pm	0.004
4	0.007	\pm	0.005	4	0.130	\pm	0.004	4	0.070	\pm	0.004
5	0.002	\pm	0.005	5	0.071	\pm	0.004	5	0.028	\pm	0.004
6	0.009	\pm	0.005	6	0.043	\pm	0.004	6	0.022	\pm	0.004
7	0.011	\pm	0.005	7	0.023	\pm	0.004				
8	-0.002	\pm	0.005	8	0.005	\pm	0.004				
9	-0.009	\pm	0.005								

Tabelle B.2: Wertetabellen für die Messungen der Wilson-Loops bei $\beta = 2.3$. Nach 1000 Sweeps zur Equilibrierung wurden die Messungen jeweils mit 5 sweeps Abstand genommen, um Korrelation zu reduzieren.

Literatur

- [1] H. Fritzsch, *The history of QCD*, CERN Courier, Sept **27** (2012) (siehe S. [1](#)).
- [2] C. DeTar und S. Gottlieb, *Lattice quantum chromodynamics comes of age*, Physics Today **57** (2004) 45 (siehe S. [1](#)).
- [3] M. Creutz und B. Freedman, *A statistical approach to quantum mechanics*, Annals of Physics **132** (1981) 427 (siehe S. [3–5](#), [8](#)).
- [4] C. Gattringer und C. Lang, *Quantum chromodynamics on the lattice: an introductory presentation*, Bd. 788, Springer Science & Business Media, 2009 (siehe S. [9–13](#), [15](#)).
- [5] M. Petschlies und C. Urbach, *Computational Physics*, 2020 (siehe S. [11](#), [17](#), [18](#)).
- [6] G. P. Lepage, *Lattice QCD for novices*, Strong Interactions at Low and Intermediate Energies (1998) 49 (siehe S. [14](#)).
- [7] H. J. Rothe, *Lattice gauge theories: an introduction*, Bd. 74, World Scientific Publishing Company, 2005 (siehe S. [15](#)).

Abbildungsverzeichnis

2.1	Ein Beispiel für eine Trajektorie des harmonischen Oszillators nach 10000 sweeps des Metropolisalgorithmus'.	7
2.2	Die gemessenen Energien \overline{E} als Funktion der Monte-Carlo-Zeit. Es ist klar eine Konvergenz zu einem Plateau bei ≈ 0.5 erkennbar.	8
2.3	Zustandsdichte des Grundzustands des harmonischen Oszillators.	9
4.1	Die durchschnittliche Plaquette eines 16×8^3 -Gitters als Funktion der Iterationszahl: nach ca. 5 sweeps scheint das System bereits einen Gleichgewichtszustand zu erreichen.	23
4.2	Links: Alle Messwerte eines 1×3 -Wilson-Loops. Rechts: Verteilung relevanter Messwerte (nach Equilibrierung alle 5 sweeps gemessen).	24
4.3	Beispiel für Wilson-Loop-Messwerte zur Bestimmung des statischen Potentials für $r = 3a$. Verwendetes Modell: $\overline{W}(3a, t) = A \cdot \exp(-aVt/a)$	25
4.4	Das statische Potential in Abhängigkeit vom Abstand von Quark und Antiquark: es ist klar ein linearer Zusammenhang erkennbar.	26
A.1	Messwerte der Wilson-Loops bei $\beta = 2.3$ und festgehaltenem r . Verwendetes Modell: $\overline{W}(r, t) = A \cdot \exp(-aV(r)t/a)$	29
A.2	Messwerte der Wilson-Loops bei $\beta = 2.3$ und festgehaltenem r . Verwendetes Modell: $\overline{W}(r, t) = A \cdot \exp(-aV(r)t/a)$	30