



Institut für Theoretische Physik

Bachelorarbeit

Computation of the static quark-antiquark-potential using non-string-like trial states

Eingereicht am 14.09.2015

von

Tobias Neitzel

Betreuer, Erstgutachter: Prof. Dr. Marc Wagner

Zweitgutachter: Prof. Dr. Owe Philipsen

Inhaltsverzeichnis

1	Einleitung	3
2	Theoretische Grundlagen	4
2.1	Übergangsamplituden	4
2.2	Wick-Rotation	4
2.3	Vakuumerwartungswerte	5
2.4	Erzeugende Funktionale	5
2.5	Diskretisierung der Raumzeit	6
2.6	Gitter Eichtheorie	6
2.7	Das Schwinger Linienintegral	7
2.8	Monte-Carlo-Simulationen	8
3	Das statische Quark-Antiquark-Potential	8
3.1	Berechnung des statischen Quark-Antiquark-Potentials mittels Wilson Loops	8
3.2	Vorstellung des neuen Verfahrens	11
3.3	Temporale Eichung	14
3.4	Off-Axis Probleme	14
3.5	Smearing-Verfahren	15
3.6	Laufzeitverhalten	15
4	Vorstellung der Ergebnisse	16
4.1	Ergebnisse unter Verwendung von string-like trial states	16
4.2	Ergebnisse unter Verwendung von non-string-like trial states	17
4.3	Ergebnisse für off-axis Abstände	20
4.4	Laufzeitverhalten	21
5	Diskussion der Ergebnisse	24
6	Programm Dokumentation	25
6.1	Grundlegende-Programme	25
6.2	Smearing	25
6.3	Berechnung der Eigenvektoren	25
6.4	Berechnung des Wilson Loop	26
6.5	Programmtests	26
6.6	Speicherverwaltung	26
6.7	Verwendung des Programms	27

1 Einleitung

In dieser Arbeit soll ein neues Verfahren zur Berechnung des statischen Quark-Antiquark-Potentials vorgestellt werden. Die zugrunde liegende Theorie ist eine $SU(2)$ -Yang-Mills-Theorie, also im wesentlichen eine zweifarbige QCD mit unendlich schweren Quarks. Die Idee hinter dem neuen Verfahren ist die Einführung von non-string-like trial states, welche numerisch sehr einfach zu handhaben sind. Ziel ist es, mit dieser Methode schnellere Laufzeiten zu erreichen und einen einfacheren Zugang zu Problemen wie off-axis Abständen zu ermöglichen.

2 Theoretische Grundlagen

Im Folgenden wird intensiv mit dem Pfadintegralformalismus gearbeitet. Der Pfadintegralformalismus bietet eine äußerst elegante Möglichkeit der Feldquantisierung und liefert numerischen Zugang zu vielen physikalisch relevanten Rechnungen. In diesem Kapitel sollen einige Grundzüge zusammengefasst werden, um einen groben Überblick zu bieten.

2.1 Übergangsamplituden

Die erste wichtige Erkenntnis ist, dass sich Übergangsamplituden durch Pfadintegrale ausdrücken lassen. Im Folgenden wird das Heisenberg Bild verwendet, also zeitunabhängige Zustände und zeitabhängige Operatoren. Da der Entwicklungspunkt für Zustände aber häufig nicht der Gleiche ist, wird folgende Notation verwendet:

$$|\phi, t\rangle = e^{i\hat{H}t}|\phi\rangle \quad (2.1)$$

$$\hat{\phi}(x) = \hat{\phi}(\mathbf{x}, t) = e^{i\hat{H}t}\hat{\phi}(\mathbf{x})e^{-i\hat{H}t} \quad (2.2)$$

Diese Darstellung sorgt dafür, dass Zustände immer korrekt zum Zeitpunkt des Operators entwickelt werden. Im Folgenden werden wir die Hüte bei Operatoren jedoch auslassen. Ein Beispiel für eine Übergangsamplitude ist der Übergang eines Feldeigenzustandes $|\phi\rangle$ zum Zeitpunkt t_0 zu einem Feldeigenzustand $|\phi'\rangle$ zum Zeitpunkt t_1 . Diese lässt sich mit der hier verwendeten Notation ausdrücken durch:

$$G(\phi', t_1, \phi, t_0) = \langle \phi', t_1 | \phi, t_0 \rangle \quad (2.3)$$

$$= \langle \phi' | e^{-iH(t_1-t_0)} | \phi \rangle \quad (2.4)$$

Solche Übergangsamplituden werden auch als Green-Funktion bezeichnet. Wie schon erwähnt, können sie auch als Pfadintegrale ausgedrückt werden. Diese haben die Form:

$$G(\phi', t_1, \phi, t_0) = \int_{\phi(\mathbf{x}, t_0)=\phi}^{\phi(\mathbf{x}, t_1)=\phi'} D\phi e^{iS[\phi]} \quad (2.5)$$

Das Integrationsmaß $D\phi$ steht dabei für die Integration über alle Feldkonfigurationen im Zeitbereich von t_0 bis t_1 . Das Feld wird also zu jeder Zeit, in diesem Intervall, und an jedem Ort über den vollständigen Wertebereich integriert. Die so entstehenden *Pfade* werden dann mit der Wirkung $S[\phi]$ gewichtet. Daher bekommt dieses Integral den Namen Pfadintegral. In dieser Form besitzen Pfadintegrale ein oszillierendes Gewicht, das für numerische Rechnungen und die physikalische Interpretation nicht brauchbar ist. Im nächsten Kapitel wird sich zeigen, dass man dieses Problem durch eine Umformulierung beheben kann.

2.2 Wick-Rotation

Wie im vorherigen Kapitel schon angedeutet, können Pfadintegrale als Integration über verschiedene Feldkonfigurationen verstanden werden, die mit dem Exponential ihrer Wirkung gewichtet werden. In der bisherigen Betrachtung oszilliert der Gewichtungsfaktor, was insbesondere für numerische Berechnungen nicht brauchbar ist. Um die Oszillation zu umgehen, kann man eine rein imaginäre Zeit verwenden. Dies entspricht einer Wick-Rotation. Man geht dann von der gewöhnlichen Zeitkomponente t über zu einer Zeitkomponente τ gegeben durch:

$$t = e^{-i\alpha}\tau \quad (2.6)$$

Die beiden gängigen Varianten für den Drehwinkel α sind $\alpha = \frac{\pi}{2}$ (Euklidische Zeit) und $\alpha \rightarrow 0$ (Minkowski Zeit). Interessanter Weise ist formal auch für Minkowski Zeit $\alpha \neq 0$, was für die Konvergenz einiger Integrale notwendig ist. Der Übergang zu einer euklidischen Zeit bringt das Pfadintegral auf die folgende Form:

$$G(\phi', t_1, \phi, t_0) = \int_{\phi(\mathbf{x}, t_0)=\phi}^{\phi(\mathbf{x}, t_1)=\phi'} D\phi e^{-S_E[\phi]} \quad (2.7)$$

Wobei $S_E[\phi]$ die Euklidische Wirkung ist. Diese muss für jede Theorie entsprechend aus der gewöhnlichen Minkowski-Wirkung abgeleitet werden. Der Gewichtungsfaktor in der Euklidischen Zeit fällt exponentiell mit steigender Wirkung ab, sodass Pfade mit geringer Wirkung stärker gewichtet werden. Dies ist eine sehr schöne Eigenschaft, da sie sich ähnlich zum Prinzip der kleinsten Wirkung verhält.

2.3 Vakuumerwartungswerte

Ein zentrales Thema der Quantenfeldtheorie ist die Berechnung von Vakuumerwartungswerten (VEV's). In diesem Kapitel wird kurz gezeigt, dass unter geeigneten Annahmen, VEV's auch über Pfadintegrale berechnet werden können. Als einfaches Beispiel betrachten wir eine Zweipunktfunktion der Feldtheorie.

$$\langle \Omega | \phi(x_1) \phi(x_2) | \Omega \rangle \quad (2.8)$$

Ziel soll es nun sein, diesen Ausdruck auf eine Übergangsamplitude zweier Feldeigenzustände zurückzuführen. Dazu betrachten wir zunächst die Übergangsamplitude mit zwei eingeschobenen Feldoperatoren, also einen Ausdruck der Form:

$$\langle \phi', \frac{\tau}{2} | \phi(x_1) \phi(x_2) | \phi, -\frac{\tau}{2} \rangle \quad (2.9)$$

Durch das Einschleiben einer $\mathbb{1}$ in Form von Energieeigenzuständen, kann dieser Ausdruck umgeschrieben werden zu:

$$\sum_{n,m} \langle \phi' | E_n \rangle \langle E_m | \phi \rangle \langle E_n | \phi(x_1) \phi(x_2) | E_m \rangle e^{-E_m \frac{\tau}{2}} e^{-E_n \frac{\tau}{2}} \quad (2.10)$$

Wird nun der Grenzwert $\tau \rightarrow \infty$ gebildet, wird die Summe in (2.10) vom Zustand mit niedrigster Energie dominiert. Dieser Zustand entspricht in der Feldtheorie gerade dem Vakuum $|\Omega\rangle$. Es bleibt also:

$$\langle \phi', \frac{\tau}{2} | \phi(x_1) \phi(x_2) | \phi, -\frac{\tau}{2} \rangle \xrightarrow{\tau \rightarrow \infty} \langle \phi' | \Omega \rangle \langle \Omega | \phi \rangle \langle \Omega | \phi(x_1) \phi(x_2) | \Omega \rangle e^{-E_\Omega \tau} \quad (2.11)$$

Wie man nun sieht, enthält Gleichung (2.11) den gesuchten Vakuumerwartungswert. Die Vorfaktoren können eliminiert werden, indem man durch die Übergangsamplitude, ohne eingeschobene Feldoperatoren, teilt. Es ergibt sich:

$$\frac{\langle \phi', \frac{\tau}{2} | \phi(x_1) \phi(x_2) | \phi, -\frac{\tau}{2} \rangle}{\langle \phi', \frac{\tau}{2} | \phi, -\frac{\tau}{2} \rangle} \xrightarrow{\tau \rightarrow \infty} \langle \Omega | \phi(x_1) \phi(x_2) | \Omega \rangle \quad (2.12)$$

Somit kann der Vakuumerwartungswert wieder durch Übergangsamplituden ausgedrückt werden. Die Feldeigenzustände ϕ und ϕ' können an sich frei gewählt werden. Lediglich der Überlapp zum Vakuum darf nicht verschwinden: $\langle \phi'(\mathbf{x}) | \Omega \rangle \neq 0$ und $\langle \Omega | \phi(\mathbf{x}) \rangle \neq 0$, sonst würde der Grundzustand auch für den gewählten Limes nicht dominieren.

Letztendlich bleibt zu zeigen, dass Gleichung (2.12) durch ein Pfadintegral ausgedrückt werden kann. Für den Nenner ist dies bereits aus Kapitel (2.1) bekannt. Auch für den Zähler lässt sich eine Pfadintegral-Darstellung finden:

$$\langle \phi', \frac{\tau}{2} | T \{ \phi(x_1) \phi(x_2) \} | \phi, -\frac{\tau}{2} \rangle = \int D\phi \phi(x_1) \phi(x_2) e^{-S_E[\phi]} \quad (2.13)$$

Dabei steht T für den Zeitordnungsoperator. Dieser ist für die Herleitung von (2.13) essenziell, da manche Pfade sonst doppelt belegt wären. Die Integrationsgrenzen sind analog zu Gleichung (2.5) zu wählen. Sie werden im Folgenden jedoch stets weggelassen.

Um den Vakuumerwartungswert also auf eine durch Pfadintegrale auswertbare Form zu bringen, schreiben wir:

$$\frac{\langle \phi', \frac{\tau}{2} | T \{ \phi(x_1) \phi(x_2) \} | \phi, -\frac{\tau}{2} \rangle}{\langle \phi', \frac{\tau}{2} | \phi, -\frac{\tau}{2} \rangle} \xrightarrow{\tau \rightarrow \infty} \langle \Omega | T \{ \phi(x_1) \phi(x_2) \} | \Omega \rangle \quad (2.14)$$

2.4 Erzeugende Funktionale

Wie im vorherigen Kapitel gezeigt wurde, können Vakuumerwartungswerte auf Pfadintegrale zurückgeführt werden. Im Fall einer quadratischen Wirkung lassen sich diese Pfadintegrale sogar analytisch behandeln. Dies geschieht durch die Verwendung sogenannter erzeugender Funktionale. Diese Objekte sollen hier kurz vorgestellt werden.

Ziel sei die Berechnung eines Pfadintegrals der Form:

$$P(x_1, \dots, x_n) = \int D\phi \phi(x_1) \dots \phi(x_n) e^{-S_E[\phi]} \quad (2.15)$$

Um dieses Integral zu Berechnen, wird das erzeugende Funktional $Z[j]$ eingeführt:

$$Z[j] = \int D\phi e^{-S_E[\phi] + \int dx^4 J(x) \phi(x)} \quad (2.16)$$

Das Pfadintegral $P(x_1, \dots, x_n)$ kann nun aus der Funktionalableitung des Erzeugenden Funktionals gewonnen werden.

$$P(x_1, \dots, x_n) = \frac{\delta^n Z[j]}{\delta J(x_1) \dots \delta J(x_n)} \Big|_{j=0} \quad (2.17)$$

Die erfolgreiche Berechnung eines erzeugenden Funktionals liefert also gleichzeitig die Lösung des Pfadintegrals. Wie oben schon erwähnt, ist die analytische Berechnung für quadratische Wirkungen möglich. Dies betrifft insbesondere freie Feldtheorien. Die Wirkung kann dann auf die folgende Form gebracht werden:

$$S[\phi] = -\frac{1}{2} \int dy^4 \int dx^4 \phi(x) A(x, y) \phi(y) \quad (2.18)$$

Diese Form ist sehr angenehm zur Berechnung des erzeugenden Funktionals. Es lässt sich zeigen, dass jedes Funktional mit einer solchen Wirkung folgende Lösung besitzt:

$$Z[j] = \zeta e^{\frac{1}{2} \int dx^4 \int dy^4 J(x) A^{-1}(x, y) J(y)} \quad (2.19)$$

Dabei bezeichnet ζ eine Konstante, die im Falle von unendlich vielen Freiheitsgraden divergiert. Diese Divergenz ist jedoch nicht weiter störend, da ζ bei der Berechnung von Vakuumerwartungswerten gerade verschwindet. Von besonderer Bedeutung ist die Matrix A^{-1} . Sie kann als Propagator der entsprechenden Theorie verstanden werden, da sie gerade das Ergebnis der Zweipunktfunktion ist.

Insgesamt sind nun alle Möglichkeiten gegeben, um n-Punkt-Funktionen mit beliebiger quadratischer Wirkung zu berechnen. Die n-Punkt-Funktionen schreiben wir im folgenden als $G_n(x_1, \dots, x_n)$.

$$G_n(x_1, \dots, x_n) = \frac{1}{\zeta} \frac{\delta^n Z[j]}{\delta J(x_1) \dots \delta J(x_n)} \Big|_{j=0} \quad (2.20)$$

2.5 Diskretisierung der Raumzeit

Bisher wurde gezeigt, dass physikalische Observablen als Pfadintegrale ausgedrückt werden können. Die so notierten Integrale sind bisher allerdings nur Formal definiert. Dies liegt daran, dass aufgrund der Kontinuität der Raumzeit diese Integrale unendlichdimensional und somit nicht zu lösen sind. Um sie also zu auswertbaren Objekten zu machen, ist ein Übergang zu einer diskreten Raumzeit erforderlich. Dies wird durch Einführung eines Raumzeit-Gitters mit fixem Gitterabstand realisiert. Zusätzlich wird dieses Gitter auf eine endliche Ausdehnung beschränkt und es werden periodische Randbedingungen angenommen. Dies sollte dem Leser ständig gegenwärtig sein, da auf dem endlichen Gitter Diskretisierungsfehler (discretization errors) und Periodizitätsfehler (finite-size-errors) auftreten.

2.6 Gitter Eichtheorie

Eichtheorien sind Feldtheorien, die eine lokale Eichsymmetrie besitzen. Es ist eine große Errungenschaft der Physik, diese Symmetrien mit den Fundamentalkräften in Beziehung zu setzen. Im Folgenden soll in aller Kürze beschrieben werden, wie man diese Theorien aus dem Eichprinzip gewinnt, und wie sich die Vorgehensweise auf dem Kontinuum und dem Gitter unterscheidet. Betrachtet wird dazu die Lagrangedichte des freien Dirac-Feldes gegeben durch:

$$\mathcal{L}[\psi, \bar{\psi}] = \bar{\psi} (i\gamma^\mu \partial_\mu - m) \psi \quad (2.21)$$

Wobei der Dirac-Spinor ψ ein N-komponentiger Vektor im Farbraum sein soll. Nun stellt man die Forderung, dass diese Lagrangedichte invariant unter lokalen $SU(N)$ Transformationen sein soll, welche im Folgenden mit $G(x)$ bezeichnet werden. Offensichtlich ist diese Invarianz von alleine nicht gegeben, was an der Ableitung in (2.21) liegt. Um Invarianz zu erreichen, müssen neue Felder A_μ^t eingeführt werden, welche als Eichfelder bezeichnet werden. Die Anzahl der nötigen Eichfelder hängt von der Symmetriegruppe ab und entspricht der Anzahl an Generatoren Θ der Gruppe. Weiterhin bezeichnet das Eichfeld, ohne oberen Index, die Summe der Eichfelder multipliziert mit den entsprechenden Generatoren $A_\mu = A_\mu^t \Theta_t$ und besitzt folgendes Transformationsverhalten unter Eichtransformationen:

$$A_\mu(x) \rightarrow A'_\mu(x) = G(x) \left(A_\mu(x) - \frac{i}{g} \partial_\mu \right) G^{-1}(x) \quad (2.22)$$

Wobei g die Kopplungskonstante der entsprechenden Theorie ist. Mithilfe dieser Definitionen lässt sich nun eine kovariante Ableitung $D_\mu = \partial_\mu + igA_\mu$ definieren, welche das Transformationsverhalten

$$D_\mu \rightarrow D'_\mu = G(x) D_\mu G^{-1}(x) \quad (2.23)$$

unter den Eichtransformationen $G(x)$ besitzt. Es ist leicht nachzurechnen, dass unter Verwendung der kovarianten Ableitung in (2.21) der Ausdruck eichinvariant wird. Die somit eingeführten Eichfelder entsprechen dann den Eichbosonen der Theorie.

$$\mathcal{L}[\psi, \bar{\psi}, A] = \bar{\psi} (i\gamma^\mu D_\mu - m) \psi \quad (2.24)$$

Die so eingeführte Lagrangedichte berücksichtigt allerdings nur Wechselwirkungen zwischen Dirac- und Eichfeldern. Für eine sinnvolle Theorie müssen noch dynamische Terme der Eichfelder konstruiert werden. Diese Ausdrücke müssen wiederum eine Eichinvarianz, sowie eine Lorentzinvarianz aufweisen, was die Wahl sehr einschränkt. Als geeignete Wahl ergibt sich:

$$\mathcal{L}[\psi, \bar{\psi}, A] = \bar{\psi} (i\gamma^\mu D_\mu - m) \psi - \frac{1}{2} \text{Tr} (F^{\mu\nu} F_{\mu\nu}) \quad (2.25)$$

$$F_{\mu\nu} = \frac{i}{g} [D_\mu, D_\nu] \quad (2.26)$$

Dies ist im Großen und Ganzen die Eichtheorie im Kontinuum. Leider stellt sich heraus, dass diese Vorgehensweise nicht eins zu eins auf das Gitter übertragbar ist. Die Eichinvarianz wird nämlich durch die Diskretisierung gebrochen. Es lassen sich jedoch andere Wege finden, um eine Eichinvarianz auf dem Gitter zu erreichen. Dazu werden jedoch neue Objekte benötigt, welche dann in der kovarianten Ableitung auf dem Gitter auftauchen. Diese Objekte werden im nächsten Abschnitt vorgestellt.

2.7 Das Schwinger Linienintegral

Das Schwinger Linienintegral bietet eine einfache Möglichkeit unter lokalen Eichtransformationen invariante Ausdrücke zu konstruieren. Diese Invarianz hält insbesondere auf dem Gitter und geht für eine verschwindende Gitterkonstante in die kovariante Ableitung des Kontinuums über. Die einfachste Form besitzt das Schwinger Integral für eine lokale $U(1)$ Symmetrie. In diesem Fall kann es gegeben werden durch:

$$U(x, y) = e^{ie \int_x^y dz^\mu A_\mu(z)} \quad (2.27)$$

Dabei ist e die Kopplungskonstante der Symmetriegruppe $U(1)$. Das Integral wird über einen Pfad von x nach y berechnet. Innerhalb des Integrals ist über μ zu summieren. Wobei sich das Eichfeld gemäß (2.22) transformiert. Das Transformationsverhalten des Integrals ist nun gegeben durch:

$$U(x, y) \rightarrow G(x)U(x, y)G^\dagger(y) \quad (2.28)$$

Dies lässt sich für eine $U(1)$ Symmetrie leicht beweisen:

$$U(x, y) \rightarrow e^{ie \int_x^y dz^\mu (A_\mu(z) - \frac{1}{e} \partial_\mu \Lambda(z))} \quad (2.29)$$

$$= e^{ie \int_x^y dz^\mu A_\mu(z) - i \int_0^1 \partial_\mu \Lambda(z(s)) \frac{dz^\mu}{ds} ds} \quad (2.30)$$

$$= e^{ie \int_x^y dz^\mu A_\mu(z) - i \int_0^1 \frac{\partial}{\partial s} \Lambda(z(s)) ds} \quad (2.31)$$

$$= e^{ie \int_x^y dz^\mu A_\mu(z) - i\Lambda(y) + i\Lambda(x)} \quad (2.32)$$

$$= G(x)U(x, y)G^\dagger(y) \quad \square \quad (2.33)$$

Die Verallgemeinerung auf $SU(N)$ ist überschaubar. Zum einen muss man von einem Eichfeld auf die entsprechende Anzahl der Gruppengeneratoren übergehen, zum anderen muss vor dem Schwinger Integral noch der sogenannte path-ordering-Operator P ergänzt werden. Das path-ordering ist so zu verstehen, dass die Matrizen in der Reihenentwicklung der Exponentialfunktion nach der Größe ihres Laufindex sortiert werden. Beschreibt der Parameter s also den Integrationsweg $z(s)$ von x nach y , mit $z(0) = y$ und $z(1) = x$, dann werden die Matrizen in jedem Term so sortiert, dass höhere Werte von s links stehen.

$$U(x, y) = P \left\{ e^{ig \int_x^y dz^\mu \tilde{A}_\mu(z)} \right\} \quad (2.34)$$

Wobei $\tilde{A}_\mu(z)$ die Eichfelder der entsprechenden Gruppe beschreibt: $\tilde{A}_\mu(z) = A_\mu^t(z)\Theta_t$, mit den Generatoren der Gruppe Θ_t . Auf den expliziten Beweis der Transformationseigenschaft aus Gleichung (2.33) wird hier verzichtet, man findet ihn zum Beispiel in [1]. Die Idee ist, die Analogie zum Zeitentwicklungsoperator zu nutzen, wodurch sich auch die Notwendigkeit des path-orderings ergibt.

Natürlich kann jedes endliche Linienintegral durch Verknüpfung mehrerer kleiner Linienintegrale ausgedrückt werden. Diese Eigenschaft ist besonders auf dem Gitter nützlich. Dort werden wir endliche Linienintegrale als Multiplikation von Links ausdrücken. Links sind dabei vom Prinzip her Linienintegrale, welche nur benachbarte Gitterpunkte miteinander verbinden.

2.8 Monte-Carlo-Simulationen

Durch die Diskretisierung der Raumzeit wird eine numerische Auswertung der vorgestellten Pfadintegrale möglich. Es ist jedoch klar, dass trotz Diskretisierung die Anzahl an auszuführenden Integrationen, selbst auf kleinen Gittern, irrsinnig groß ist. Daher ist es nötig, sich statistischer Verfahren zu bedienen.

Ein Ansatzpunkt hierfür ist der exponentielle Abfall der Pfadintegrale mit steigender Wirkung. Er sorgt dafür, dass nur ein kleiner Teil der möglichen Feldkonfigurationen überhaupt einen Beitrag zum Pfadintegral leistet. Diesen Fakt nutzt die Technik des sogenannten *importance sampling* aus. Sie erzeugt Feldkonfigurationen mit einer Wahrscheinlichkeit, welche gerade diesem exponentiellen Gewichtungsfaktor entspricht. Wird auf diese Weise eine repräsentative Menge an Konfigurationen erzeugt, ergibt sich das Pfadintegral als einfacher Mittelwert des Integranden über die entsprechenden Konfigurationen.

Natürlich ist zu beachten, dass durch diese Prozedur statistische Fehler entstehen. Für gute Ergebnisse ist es daher nötig eine ausreichende Anzahl an Konfigurationen zu verwenden und jede Konfiguration maximal auszuschöpfen.

3 Das statische Quark-Antiquark-Potential

Im Folgenden soll eine ungefähre Vorstellung davon gegeben werden, wie man das Potential zwischen zwei statischen (unendlich schweren) Quarks berechnen kann. Zunächst wird die traditionelle Berechnung mit Hilfe von Wilson Loops betrachtet. Danach wird das neue Verfahren vorgestellt, welches die Eigenwerte des kovarianten Laplaceoperators verwendet. Für beide Fälle betrachten wir eine $SU(2)$ Eichsymmetrie.

3.1 Berechnung des statischen Quark-Antiquark-Potentials mittels Wilson Loops

Für die Berechnung des statischen Quark-Antiquark-Potentials benötigt man zunächst eine Art Test-Zustand (*trial state*), welcher zwei räumlich separierte, unendlich schwere Quarks beinhaltet. Dies kann durch die Anwendung der entsprechenden Feldoperatoren auf das Vakuum erreicht werden. Damit der Zustand physikalisch sinnvoll ist, muss er zudem eichinvariant sein. Wie im vorherigen Kapitel gezeigt, erreicht man dies durch den Einsatz von Schwinger Linienintegralen. Ein möglicher Zustand wäre also:

$$|\Phi_{\mu\nu}(\mathbf{x}, \mathbf{y}, 0)\rangle = \bar{Q}_\mu(\mathbf{x}, 0)U(\mathbf{x}, 0, \mathbf{y}, 0)Q_\nu(\mathbf{y}, 0)|\Omega\rangle \quad (3.1)$$

Mit einem Quark an Position $(\mathbf{x}, 0)$ und einem Antiquark bei $(\mathbf{y}, 0)$. Die Indizes μ und ν entsprechen den Spinindizes der Quarks. Zusätzlich sind die Quarks zweikomponentige Vektoren im Farbraum. Das Linienintegral steht inmitten der beiden Feldoperatoren und entspricht einer 2×2 -Matrix im Farbraum. Wir sprechen in diesem Zusammenhang von einem string-like trial state, da es die beiden Quarks wie eine Linie miteinander verbindet.

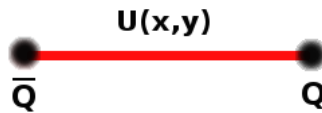


Abbildung 1: string-like trial state

Es wäre nun naiv zu erwarten, dass der so konstruierte Zustand direkt ein Energieeigenzustand zum entsprechenden Hamiltonian ist. Deswegen trägt er auch nur die Bezeichnung trial state, und stellt den Versuch dar, dem von der Natur realisierten Zustand möglichst nahe zu kommen. Dies ermöglicht nämlich die Berechnung der Energie, des eigentlich realisierten Zustandes. Dies funktioniert nach dem Prinzip, welches in Kapitel (2.3) vorgestellt wurde. Dort wurde gezeigt, dass sich eine Übergangsamplitude als eine, mit der Zeit abfallende, Exponentialfunktion darstellen lässt, wobei der Abfall gerade proportional zur Energie der Energieeigenzustände ist. Da für große Zeiten die niedrigste Energie dominiert, erlaubt die Berechnung der Übergangsamplitude Rückschlüsse auf die Energie des niedrigsten Energieeigenzustandes. Allerdings ist zu beachten, dass wir hier keine Feldeigenzustände betrachten. Daher unterscheidet sich die Vorgehensweise etwas von der in Kapitel (2.3). Dies führt unter anderem dazu, dass die Vakuumenergie nicht zum exponentiellen Abfall beiträgt, was für unsere Zwecke natürlich sehr nützlich ist.

Wir konstruieren also einen zweiten trial state zu einer späteren Zeit t und setzen die Quarks diesmal an die Positionen x' und y' . Ziel ist nun die Berechnung der Übergangsamplitude. Mit Hilfe von Kapitel (2.3) und ein wenig physikalischer Interpretation können wir die ungefähre Form des Ergebnisses schon voraussagen:

$$\langle \Phi_{\mu'\nu'}(\mathbf{x}', \mathbf{y}', t) | \Phi_{\mu\nu}(\mathbf{x}, \mathbf{y}, 0) \rangle \xrightarrow[M_Q \rightarrow \infty]{t \rightarrow \infty} \delta(\mathbf{x}' - \mathbf{x})\delta(\mathbf{y}' - \mathbf{y})C_{\mu'\nu'\mu\nu}(\mathbf{x}, \mathbf{y})e^{-E(R)t} \quad (3.2)$$

Die Deltafunktionen kommen durch die unendliche Masse der Quarks und sorgen dafür, dass die unendlich schweren Quarks nicht propagieren können. Der Faktor C beschreibt den Überlapp zwischen dem zuvor konstruierten trial state und dem niedrigsten Energieeigenzustand des Hamiltonian. Der exponentielle Abfall geht entsprechend mit der Energie dieses Energieeigenzustandes abzüglich der Vakuumenergie. Für die explizite Berechnung der Übergangsamplitude ist es zweckmäßig diese zunächst komplett auszusprechen:

$$\langle \Omega | \bar{Q}_{\nu'}(\mathbf{y}', t) U(\mathbf{y}', t, \mathbf{x}', t) Q_{\mu'}(\mathbf{x}', t) \bar{Q}_{\mu}(\mathbf{x}, 0) U(\mathbf{x}, 0, \mathbf{y}, 0) Q_{\nu}(\mathbf{y}, 0) | \Omega \rangle \quad (3.3)$$

In diesem Ausdruck müssen nun einige Feldoperatoren vertauscht werden, dabei ist die mathematische Struktur des Ausdrucks im Farbraum natürlich beizubehalten. Das Ergebnis ist momentan ein Skalar im Farbraum und muss dies auch bleiben. Daher muss bei einer Vertauschung stets auf die Matrix-Vektor-Struktur geachtet werden. Weiterhin ist zu beachten, dass die Quarks durch Grassmann-Variablen beschrieben werden, somit ändert die Vertauschung zweier Operatoren das Vorzeichen. Dies führt insgesamt dazu, dass im folgenden Ausdruck die Spur auftaucht und ein zusätzliches negatives Vorzeichen vorhanden ist.

$$- \langle \Omega | Tr \left(U(\mathbf{y}', t, \mathbf{x}', t) Q_{\mu'}(\mathbf{x}', t) \bar{Q}_{\mu}(\mathbf{x}, 0) U(\mathbf{x}, 0, \mathbf{y}, 0) Q_{\nu}(\mathbf{y}, 0) \bar{Q}_{\nu'}(\mathbf{y}', t) \right) | \Omega \rangle \quad (3.4)$$

Da es sich hierbei wieder um einen Vakuumerwartungswert handelt, kann man diesen Ausdruck ebenso als Pfadintegral schreiben. Wie in Kapitel (2.4) angedeutet, gehen die Pärchen von Quark-Operatoren dann in Propagatoren über. Dabei ist zu beachten, dass wir eine Eichtheorie betrachten. Somit ist die Wirkung nicht frei und es ergeben sich Propagatoren im externen Feld A_{μ} . Die Tatsache, dass wir nur unendlich schwere Quarks betrachten, vereinfacht die Rechnung an dieser Stelle wesentlich. Für endliche Massen würden zusätzliche Vakuum-Polarisationseffekte hinzuommen [2]. In unserem Fall ergibt sich jedoch einfach die Ersetzung:

$$Q_{\mu'}(\mathbf{x}', t) \bar{Q}_{\mu}(\mathbf{x}, 0) \rightarrow S_{\mu'\mu}(\mathbf{x}', t, \mathbf{x}, 0, A) \quad (3.5)$$

Wobei der Propagator S im externen Feld A_{μ} für unendlich schwere Quarks gegeben ist durch:

$$S_{\mu'\mu}(\mathbf{x}', t, \mathbf{x}, 0, A) \stackrel{M_Q \rightarrow \infty}{=} \delta(\mathbf{x}' - \mathbf{x}) U(\mathbf{x}, t, \mathbf{x}, 0) C_{\mu'\mu} e^{-M_Q t} \quad (3.6)$$

Die Deltafunktion war dabei zu erwarten, da die Quarks wegen der unendlich schweren Masse räumlich nicht propagieren dürfen. Die Abhängigkeit vom Eichfeld erzeugt das zeitliche Linienintegral U , welches an der räumlichen Position \mathbf{x} die Zeitdifferenz der beiden Zustände durchläuft. Die Spin-Struktur des Ausdrucks kann durch eine Konstante C zusammengefasst werden. Besonders wichtig für die spätere Rechnung ist der exponentielle Abfall mit der Quark-Masse.

In gleicher Weise kann auch für die Quark-Operatoren bei y und y' vorgegangen werden. Diese liefern einen analogen Ausdruck, welcher aufgrund der umgekehrten Zeitstruktur ein zeitliches Linienintegral mit entgegengesetzter Laufrichtung zum Vorherigen besitzt. Der Vollständigkeit halber muss an dieser Stelle noch erwähnt werden, dass die Kombinatorik auch Propagatoren von x nach y und x' nach y' erzeugt. Diese spielen für die weitere Berechnung aber keine Rolle, da sie aufgrund der Deltafunktion in (3.6) keinen Beitrag leisten.

Der Übergang zu den Propagatoren entspricht im Pfadintegral-Formalismus einer Integration über die Grassmann-Variablen Q und \bar{Q} . Im Pfadintegral ist dann nur noch die Integration über die Linienintegrale U auszuführen. Somit können alle Objekte, welche nicht von den Eichfeldern abhängen, vor das verbleibende Pfadintegral geschrieben werden. Im Pfadintegral verbleibt dann lediglich die Spur aus den vier Links, welche zum einen aus der Konstruktion des trial states, zum anderen aus den Propagatoren kommen. Diese bilden eine geschlossene Kontur, welche im Folgenden als Wilson Loop bezeichnet wird.

$$W[U] = U(\mathbf{y}, t, \mathbf{x}, t) U(\mathbf{x}, t, \mathbf{x}, 0) U(\mathbf{x}, 0, \mathbf{y}, 0) U(\mathbf{y}, 0, \mathbf{y}, t) \quad (3.7)$$

Fasst man nun alle Schritte zusammen und wendet sie auf die Übergangsamplitude an, erhält man am Ende folgenden Ausdruck für die Übergangsamplitude:

$$\langle \Phi_{\mu'\nu'}(\mathbf{x}', \mathbf{y}', t) | \Phi_{\mu\nu}(\mathbf{x}, \mathbf{y}, 0) \rangle \stackrel{M_Q \rightarrow \infty}{=} \delta(\mathbf{x} - \mathbf{x}') \delta(\mathbf{y} - \mathbf{y}') \tilde{C}_{\mu'\nu\nu'} e^{-2M_Q t} \langle Tr(W[U]) \rangle \quad (3.8)$$

Die spitzen Klammern sollen dabei anzeigen, dass der eingeklammerte Ausdruck nach wie vor im Pfadintegral steht.

$$\langle W[U] \rangle = \frac{\int D U Tr(W[U]) e^{-S[U]}}{\int D U e^{-S[U]}} \quad (3.9)$$

Wie man auf den ersten Blick sieht, besitzt Gleichung (3.9) hohe Ähnlichkeit mit der prognostizierten Form. Besonders schön ist der exponentielle Abfall mit zweimal der Ruhemasse, da diese ja offensichtlich einen Teil der

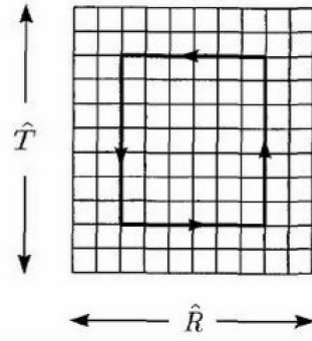


Abbildung 2: Grafische Darstellung des Wilson Loop [3]

Energie des niedrigsten Energieeigenzustandes ausmacht. Da sich dieser Teil nun quasi von selbst abgespalten hat, kann man daraus schließen, dass das Pfadintegral über den Wilson Loop, für hinreichend große Zeiten, den restlichen Energieterm, also gerade das Potential, als exponentiellen Abfall besitzt.

$$\langle W[U] \rangle \xrightarrow{t \rightarrow \infty} C(R) e^{-V(R)t} \quad (3.10)$$

Um nun also das statische Quark-Antiquark-Potential zu bestimmen, muss das Pfadintegral über den Wilson Loop ausgerechnet werden. Auf dem Gitter kann man diese Rechnung numerisch leicht durchführen. Dazu werden Eichfeldkonfigurationen mittels Monte-Carlo-Algorithmus erzeugt, welcher in Kapitel (2.8) grob erläutert wurde. Das Pfadintegral ergibt sich dann einfach, indem man den Mittelwert des Wilson Loop über die verschiedenen Konfigurationen berechnet. Der Wilson Loop selbst entspricht auf dem Gitter nur noch einer Multiplikation aus Links, welche numerisch einfach durchzuführen ist. Das Ergebnis $W(R, T)$ wird natürlich von der räumlichen Separation der Quarks und der zeitlichen Separation der Zustände abhängen. Um aus den gewonnenen Daten den exponentiellen Abfall zu extrahieren, bestimmt man dann die effektive Masse. Diese ist gegeben durch:

$$m_{eff} = \ln \left(\frac{W(R, t)}{W(R, t+1)} \right) \quad (3.11)$$

Setzt man nun für $W(R, t)$ das prognostizierte Ergebnis (3.10) ein, erhält man direkt das Potential $V(R)$.

$$m_{eff} \xrightarrow{t \rightarrow \infty} V(R) \quad (3.12)$$

Aufgrund der Grenzwertbildung ist in der numerischen Rechnung damit zu rechnen, dass die effektive Masse erst zu größeren Zeiten konvergiert. In diesem Bereich kann dann das entstehende Plateau betrachtet und das Potential somit extrahiert werden.

Ergebnisse dieser Prozedur werden in Abschnitt (4) vorgestellt. Dort werden sie insbesondere mit dem neuen Verfahren verglichen, welches im folgenden Kapitel vorgestellt wird.

3.2 Vorstellung des neuen Verfahrens

Obwohl die Berechnung des statischen Potentials mit Hilfe von Wilson Loops gute Ergebnisse liefert, wollen wir in diesem Abschnitt ein neues Verfahren untersuchen, um das Potential zu berechnen. Die Motivation dahinter baut zum einen auf einer Vereinfachung von off-axis Rechnungen, welche in Abschnitt (3.4) diskutiert werden und zum anderen auf die Verbesserung des Laufzeitverhalten des Wilson Loop. Dieser besteht nämlich, wegen der eichinvarianten Konstruktion des trial states, notwendigerweise aus räumlichen Linienintegralen, welche auf dem Gitter einer Multiplikation aus Links entsprechen.

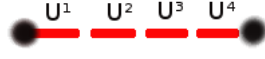


Abbildung 3: Multiplikation von Links im string-like trial state

Für größere räumliche Separationen nimmt die Anzahl der Rechnungen somit deutlich zu. Ziel ist es daher, die räumlichen Linienintegrale durch Objekte zu ersetzen, welche diesen string-artigen Charakter nicht besitzen und somit zu einem non-string-like trial state überzugehen.



Abbildung 4: Idee eines non-string-like trial state

Die Kandidaten für solche Objekte sind die Eigenvektoren des kovarianten Laplaceoperators. Dieser entspricht einer modifizierten Form des gewöhnlichen Laplaceoperators. Ähnlich wie bei der in Kapitel (2.6) vorgestellten kovarianten Ableitung werden hier Links genutzt, um folgendes Transformationsverhalten unter einer lokalen Eichtransformation $G(x)$ zu erreichen:

$$\Delta \rightarrow \Delta' = G(x)\Delta G^\dagger(x) \quad (3.13)$$

Angenommen eine Funktion $f(x)$ (im Folgenden auch als Vektor bezeichnet) sei nun Eigenvektor zum kovarianten Laplaceoperator zum Eigenwert λ . Dann erfüllt er die Eigenwertgleichung:

$$\Delta f(x) = \lambda f(x) \quad (3.14)$$

Für eine $SU(2)$ -Eichtheorie, wie sie in dieser Arbeit behandelt wird, lassen sich immer solche Eigenwerte und Eigenvektoren finden. Weiterhin zeigt sich, dass die Eigenwerte zweifach entartet sind, also zwei Eigenvektoren zu jedem Eigenwert existieren, welche die Basis eines Eigenraums bilden. Näheres kann in [3] nachgelesen werden. Von besonderem Interesse für diese Arbeit, sind die entarteten Eigenvektoren zum niedrigsten Eigenwert. Diese seien im Folgenden mit $f_1(x)$ und $f_2(x)$ bezeichnet und werden so gewählt, dass sie eine Orthonormalbasis des Eigenraums bilden.

Betrachtet man nun den kovarianten Laplaceoperator auf einer eichtransformierten Feldkonfiguration $\tilde{\Delta}$, lassen sich auch für diesen wieder zwei Eigenvektoren $\tilde{f}_{1/2}(x)$ finden.

$$\tilde{\Delta} \tilde{f}_{1/2}(x) = \lambda' \tilde{f}_{1/2}(x) \quad (3.15)$$

Da das Transformationsverhalten des kovarianten Laplaceoperators jedoch aus Gleichung (3.13) bekannt ist, kann dieser wieder durch den Laplaceoperator auf den nicht transformierten Feldkonfigurationen Δ ausgedrückt werden. Man findet dann:

$$\begin{aligned} G(x)\Delta G^\dagger(x)\tilde{f}_{1/2}(x) &= \lambda' \tilde{f}_{1/2}(x) \\ \Delta G^\dagger(x)\tilde{f}_{1/2}(x) &= \lambda' G^\dagger(x)\tilde{f}_{1/2}(x) \end{aligned} \quad (3.16)$$

Dies bedeutet aber, dass die Größe $G^\dagger(x)\tilde{f}_{1/2}(x)$ wieder Eigenvektor zum Laplaceoperator auf den nicht transformierten Feldkonfigurationen Δ ist. Da man sich außerdem darauf geeinigt hat, immer die Eigenvektoren zum niedrigsten Eigenwert zu berechnen, folgt $\lambda' = \lambda$.

Mit dieser Erkenntnis kann nun das Transformationsverhalten der Eigenvektoren angegeben werden. Jeder

Eigenvektor des nicht transformierten kovarianten Laplaceoperators Δ zum Eigenwert λ , lässt sich als Linearkombination von $f_1(x)$ und $f_2(x)$ schreiben. Aus Gleichung (3.16) ergibt sich dann:

$$\begin{aligned}\tilde{f}_1(x) &= G(x) (\alpha_1 f_1(x) + \alpha_2 f_2(x)) \\ \tilde{f}_2(x) &= G(x) (\alpha_3 f_1(x) + \alpha_4 f_2(x))\end{aligned}\quad (3.17)$$

Anhand dieses Transformationsverhaltens lässt sich nun schon vermuten, dass die Eigenvektoren als Ersatz für Linienintegrale geeignet sind. Sie müssen lediglich in geeigneter Weise multipliziert werden. Um dies zu sehen, sei hier nochmal das Transformationsverhalten eines Linienintegrals dargestellt:

$$U'(x, y) = G(x)U(x, y)G^\dagger(y) \quad (3.18)$$

Eine einfache Möglichkeit dieses Transformationsverhalten zu realisieren, bildet die folgende Kombination aus Eigenvektoren.

$$U'(x, y) \rightarrow \tilde{f}_1(x)\tilde{f}_1^\dagger(y) \quad (3.19)$$

Bei der Multiplikation handelt es sich um ein dyadisches Produkt. Da die Eigenvektoren an jedem Gitterpunkt aus zwei Farbkomponenten bestehen, entsteht so eine (2x2) Matrix im Farbraum. Diese besitzt nach Gleichung (3.17) dann das folgende Transformationsverhalten.

$$\tilde{f}_1(x)\tilde{f}_1^\dagger(y) = G(x) (\alpha_1 f_1(x) + \alpha_2 f_2(x)) (\alpha_1 f_1(y) + \alpha_2 f_2(y))^\dagger G^\dagger(y) \quad (3.20)$$

Wie man sieht, ist das Transformationsverhalten recht ähnlich zu dem in (3.18) gegebenen Linienintegral. Es stellt sich nun jedoch die Frage, ob das Auftreten der Linearkombinationen in (3.20) die Eichinvarianz zerstört. In der Tat ist dies ein Problem und die Eichinvarianz ist bei der naiv gewählten Form aus (3.19) nicht gegeben. Wegen der Linearkombinationen gilt das Transformationsverhalten aus Gleichung (3.18) nämlich nur für ein bestimmtes Paar Eigenvektoren aus den entsprechenden Eigenräumen. Bei der numerischen Berechnung der Eigenvektoren wird jedoch ein beliebiger Eigenvektor aus dem jeweiligen Eigenraum berechnet. Berechnet man also einmal die Eigenvektoren von Δ und darauf die von Δ' , besitzen diese im Allgemeinen nicht den Zusammenhang aus Gleichung (3.18).

Ziel muss es also sein, diesen Umstand zu beheben. Benötigt wird dazu eine Kombination von Eigenvektoren, welche für alle Fälle das Transformationsverhalten aus (3.18) besitzt. Es wird sich zeigen, dass folgende Kombination eine geeignete Wahl darstellt.

$$U'(x, y) \rightarrow \sum_{i=1}^2 \tilde{f}_i(x)\tilde{f}_i^\dagger(y) \quad (3.21)$$

Es ist also lediglich die Summe über die beiden Eigenvektoren zum niedrigsten Eigenwert zu bilden, um Eichinvarianz zu gewährleisten. Dabei muss allerdings vorausgesetzt werden, dass diese eine Orthonormalbasis bilden, wie wir es schon weiter oben gefordert haben. Dass diese Kombination wirklich die geforderten Eigenschaften besitzt, soll im Folgenden bewiesen werden.

Direkt ersichtlich ist, dass das neu gebildete Objekt wieder ein Transformationsverhalten der Form $G(x)...G^\dagger(y)$ aufweist. Auch hier stellt sich also nur die Frage, ob die auftretenden Linearkombinationen die Eichinvarianz zerstören. Die Linearkombinationen sind in diesem Fall gegeben durch:

$$(\alpha_1 f_1(x) + \alpha_2 f_2(x)) (\alpha_1 f_1(y) + \alpha_2 f_2(y))^\dagger + (\alpha_3 f_1(x) + \alpha_4 f_2(x)) (\alpha_3 f_1(y) + \alpha_4 f_2(y))^\dagger \quad (3.22)$$

Um genug Übersicht für die folgende Rechnung zu gewährleisten, ist es besser den Term in zwei Teile aufzuspalten:

$$(1) \rightarrow |\alpha_1|^2 f_1(x)f_1^\dagger(y) + |\alpha_2|^2 f_2(x)f_2^\dagger(y) + \alpha_1\alpha_2^* f_1(x)f_2^\dagger(y) + \alpha_2\alpha_1^* f_2(x)f_1^\dagger(y) \quad (3.23)$$

$$(2) \rightarrow |\alpha_3|^2 f_1(x)f_1^\dagger(y) + |\alpha_4|^2 f_2(x)f_2^\dagger(y) + \alpha_3\alpha_4^* f_1(x)f_2^\dagger(y) + \alpha_4\alpha_3^* f_2(x)f_1^\dagger(y) \quad (3.24)$$

Nun kann ausgenutzt werden, dass f_1 und f_2 sowie \tilde{f}_1 und \tilde{f}_2 jeweils Orthonormalbasen bilden. Wichtig ist zu beachten, dass dies nur auf dem räumlichen Vektorraum gilt. Im Farbraum dürfen die Vektoren nicht als

orthonormal angenommen werden. Aus der Orthonormalität im räumlichen Vektorraum folgt:

$$\begin{aligned} \sum_x \tilde{f}_1^\dagger(x) \tilde{f}_1(x) &= 1 \\ \rightarrow |\alpha_1|^2 + |\alpha_2|^2 &= 1 \end{aligned} \quad (3.25)$$

$$\begin{aligned} \sum_x \tilde{f}_2^\dagger(x) \tilde{f}_2(x) &= 1 \\ \rightarrow |\alpha_3|^2 + |\alpha_4|^2 &= 1 \end{aligned} \quad (3.26)$$

$$\begin{aligned} \sum_x \tilde{f}_1^\dagger(x) \tilde{f}_2(x) &= 0 \\ \rightarrow \alpha_1^* \alpha_3 &= -\alpha_2^* \alpha_4 \end{aligned} \quad (3.27)$$

Gleichung (3.27) kann mit der komplex konjugierten Gleichung multipliziert werden. Nach geeigneter Umformung lässt sich dies dann darstellen als:

$$\frac{|\alpha_1|^2}{|\alpha_2|^2} = \frac{|\alpha_4|^2}{|\alpha_3|^2} \quad (3.28)$$

Diese Gleichung kann wiederum nach dem Betragsquadrat von α_1 umgestellt werden, welches danach in Gleichung (3.25) eingesetzt werden kann.

$$|\alpha_2|^2 \frac{|\alpha_4|^2}{|\alpha_3|^2} + |\alpha_2|^2 = 1 \quad (3.29)$$

$$|\alpha_2|^2 (|\alpha_4|^2 + |\alpha_3|^2) = |\alpha_3|^2 \quad (3.30)$$

$$|\alpha_2|^2 = |\alpha_3|^2 \quad (3.31)$$

Analog lässt sich natürlich beweisen, dass dieser Zusammenhang auch für α_1 und α_4 gilt. Damit wäre der größte Teil des Beweises geschafft. Als nächstes ist Gleichung (3.27) mit α_3^* und α_2 zu multiplizieren.

$$\alpha_1^* \alpha_2 |\alpha_3|^2 = -\alpha_3^* \alpha_4 |\alpha_2|^2 \quad (3.32)$$

Da das Betragsquadrat von α_3 und α_2 nach Gleichung (3.31) jedoch gleich ist, können diese aus der Gleichung gekürzt werden.

$$\alpha_1^* \alpha_2 = -\alpha_3^* \alpha_4 \quad (3.33)$$

Nun ist man am Ende angekommen. Die Gleichungen (3.23) und (3.24) treten in (3.22) als Summe auf. Aufgrund der Relation (3.33) kürzen sich die gemischten Terme, welche sowohl f_1 als auch f_2 enthalten. Es bleibt lediglich:

$$(|\alpha_1|^2 + |\alpha_3|^2) f_1(x) f_1^\dagger(y) + (|\alpha_2|^2 + |\alpha_4|^2) f_2(x) f_2^\dagger(y) \quad (3.34)$$

Kombiniert man die Relation in (3.31) mit denen in (3.25) oder (3.26), sieht man jedoch unmittelbar:

$$|\alpha_1|^2 + |\alpha_3|^2 = 1 \quad (3.35)$$

$$|\alpha_2|^2 + |\alpha_4|^2 = 1 \quad (3.36)$$

Somit reduziert sich das Ergebnis auf die sehr einfache Form:

$$f_1(x) f_1^\dagger(y) + f_2(x) f_2^\dagger(y) \quad (3.37)$$

Insgesamt erhält man also folgendes Transformationsverhalten:

$$\sum_{i=1}^2 \tilde{f}_i(x) \tilde{f}_i^\dagger(y) = G(x) \left(\sum_{i=1}^2 f_i(x) f_i^\dagger(y) \right) G^\dagger(y) \quad (3.38)$$

Entscheidend an dieser Gleichung ist, dass keine Koeffizienten mehr vorkommen und die Basisvektoren $f_1(x)$ und $f_2(x)$ frei gewählt werden können. Dies bedeutet, dass das Transformationsverhalten aus (3.38) immer dann gilt, wenn die Eigenvektoren eine vollständige Orthonormalbasis bilden.

Zusammengefasst haben wir jetzt also einen Ersatz für die räumlichen Linienintegrale gefunden.

$$U(\mathbf{x}, t, \mathbf{y}, t) \rightarrow \sum_{i=1}^2 f_i(\mathbf{x}, t) f_i^\dagger(\mathbf{y}, t) \quad (3.39)$$

Der neue trial state ergibt sich entsprechend durch einfache Substitution der Linienintegrale.

$$|\Phi_{\mu\nu}(\mathbf{x}, \mathbf{y}, 0) \rangle = \bar{Q}_\mu(\mathbf{x}, 0) \left(\sum_{i=1}^2 f_i(\mathbf{x}, 0) f_i^\dagger(\mathbf{y}, 0) \right) Q_\nu(\mathbf{y}, 0) |\Omega \rangle \quad (3.40)$$

Der so konstruierte trial state hat nun den gewünschten non-string-like Charakter, wie er in Abbildung (3) dargestellt wurde. Für die restliche Behandlung des statischen Quark-Antiquark-Potentials verläuft alles analog zu Kapitel (3.1). Es müssen lediglich alle räumlichen Linienintegrale durch die neu gefundenen Objekte ersetzt werden. Der Wilson Loop bekommt dann die Form:

$$W[U] = \left(\sum_{i=1}^2 f_i(\mathbf{y}, t) f_i^\dagger(\mathbf{x}, t) \right) U(\mathbf{x}, t, \mathbf{x}, 0) \left(\sum_{i=1}^2 f_i(\mathbf{x}, 0) f_i^\dagger(\mathbf{y}, 0) \right) U(\mathbf{y}, 0, \mathbf{y}, t) \quad (3.41)$$

Die Veränderung des trial state schlägt sich also im Wilson Loop wieder. Dies war gerade unser Ziel. Der neue Wilson Loop spart Rechenzeit, da zwei seiner Strings durch non-string-like Objekte ersetzt wurden.

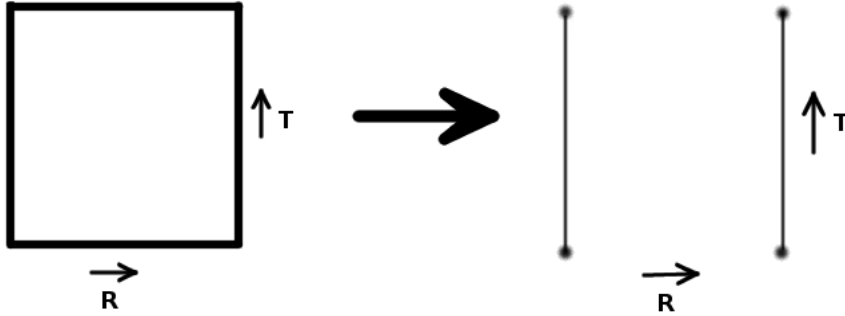


Abbildung 5: Änderung des Wilson Loops

Wie schon zuvor ist nun das Pfadintegral über den Wilson Loop auszuwerten. Über die effektive Masse können dann auch hier die entsprechenden Potentialwerte extrahiert werden. Die Ergebnisse des neuen Verfahrens werden ebenfalls in Abschnitt (4) vorgestellt.

3.3 Temporale Eichung

An dieser Stelle soll noch kurz die, auch in dieser Arbeit verwendete, temporale Eichung erklärt werden. Die temporale Eichung setzt eine Eichbedingung an die Eichfeldkonfigurationen, welche gegeben ist durch $A_0 = 0$. Auf dem Gitter interessiert man sich aber mehr für die Links, die mit dieser Eichung zu $U_0 = \mathbb{1}$ werden. Würde man es also schaffen die temporale Eichung auf dem ganzen Gitter zu implementieren, würden zusätzlich die string-artigen Verbindungen der zeitlichen Linienintegrale wegfallen.

Es zeigt sich jedoch, dass die temporale Eichung auf einem periodischen Gitter mit endlicher Gittergröße T nicht vollständig möglich ist. Sie lässt sich nur insofern realisieren, als dass ein bestimmter Link in Zeitrichtung die Eichbedingung nicht erfüllt. Dies stellt für die Berechnung der zeitlichen Linienintegrale dennoch einen großen Vorteil dar. Es muss lediglich geprüft werden, ob der Link mit $A_0 \neq 0$ in einem Linienintegral enthalten ist. Ist er das, bekommt das Linienintegral den Wert des entsprechenden Links. Ansonsten bekommt es den Wert $\mathbb{1}$. Insgesamt kann somit also auch der string-like-Charakter der zeitlichen Links umgangen werden. Wie die temporale Eichung zu implementieren ist, kann in [4] nachgelesen werden.

3.4 Off-Axis Probleme

Auf dem Gitter sind sogenannte on-axis Wilson Loops am einfachsten auszuwerten. Diese entsprechen dem Fall, dass sich die räumliche Separation der beiden Quarks auf einer Achse darstellen lässt. Wegen des diskreten Gitterabstandes ist die so erreichte Auflösung des Potentials aber auf ganzzahlige Gitterabstände beschränkt. Für eine höhere Auflösung müssen off-axis Wilson Loops berechnet werden. Diese entsprechen räumlichen Separationen, die sich auf dem Gitter nur auf zwei oder gar drei verschiedenen Achsen darstellen lassen. Somit können Abstände erzeugt werden, welche auch zwischen den Vielfachen des Gitterabstandes liegen. Die Auflösung des Potentials wird damit deutlich verbessert.

Die Implementierung dieser off-axis Abstände erweist sich mit gewöhnlichen Wilson Loops als schwierig. Grund dafür ist, dass es keine Links gibt, welche schräg durch das Gitter verlaufen. Um zwei off-axis separierte Quarks

durch ein Linienintegral zu verbinden, müssen daher komplizierte, treppenartige Wege mit Links gegangen werden.

Dieses Problem wird von dem in (3.2) neu vorgestellten Verfahren vollständig umgangen. Dadurch, dass die Quarks nicht mehr durch string-artige Objekte verknüpft sind, ist die Berechnung von off-axis Separationen sehr einfach und anschaulich zu implementieren. Dies stellt einen weiteren, wenn nicht sogar den großen Vorteil des neuen Verfahrens dar. In Abschnitt (4) werden die Ergebnisse für solche off-axis Separationen vorgestellt.

3.5 Smearing-Verfahren

Für die in diesem Kapitel vorgestellten Verfahren, ist es essenziell, dass die konstruierten trial states einen Überlapp zum niedrigsten Energieeigenzustand besitzen. Mit dem Wissen aus Kapitel (2.3) ist es auch anschaulich, dass sich die Ergebnisse verbessern, je größer dieser Überlapp ist. Ein höherer Überlapp bedeutet nämlich eine frühere Unterdrückung der angeregten Zustände und somit eine schnellere Konvergenz der effektiven Masse an die gewünschten Plateaus.

Um den Überlapp zum Grundzustand zu erhöhen, wurde in dieser Arbeit ein so genanntes smearing-Verfahren verwendet. Dieses unterteilt sich nochmal in das APE-smearing und das HYP-smearing. Grob gesagt, sorgen beide Verfahren für ein *verschmieren* der Links, wobei sich das APE-smearing auf räumliche, das HYP-smearing auf zeitliche Links auswirkt. Dazu wird jeder Link durch eine Schlaufe über benachbarte Links ergänzt, welche dann mit einem Gewichtungsfaktor zum geschmierten Link beitragen. Im Fall des string-like trial state, kann man sich das APE-smearing sehr anschaulich als Verbreiterung der Verbindungslinie vorstellen. Dadurch wird der verbindende string quasi zu einem Schlauch, welcher die beiden Quarks verbindet. Dies sorgt für eine deutliche Verbesserung der Ergebnisse. Für den non-string-like trial state ist diese anschauliche Interpretation nicht möglich. Trotzdem zeigt sich, dass auch hier das APE-smearing zu einer Verbesserung der Ergebnisse führt.

3.6 Laufzeitverhalten

In diesem Abschnitt soll das Laufzeitverhalten der Verfahren analysiert werden, denn schließlich war dies eine Motivation für das neue Verfahren. Dazu betrachten wir jeweils eine Konfiguration auf einem quadratischen Gitter (Gittergröße: N) und überlegen uns das Laufzeitverhalten in Abhängigkeit von der Gittergröße. Während für das alte Verfahren nur die Laufzeit des Wilson Loops eine Rolle spielt, muss für das neue Verfahren auch die Laufzeit des Eigenvektorrechners betrachtet werden.

- **Eigenvektorrechner**

Der Eigenvektorrechner wurde aus einer anderen Arbeit übernommen [3] und verwendet zudem das, bezüglich der Laufzeit, recht undurchsichtige Arnoldi-Package (ARPACK). Daher kann für das Laufzeitverhalten dieses Programmteils keine theoretische Aussage getroffen werden.

- **Der gewöhnliche Wilson Loop**

Der Code für den gewöhnlichen Wilson Loop nutzt zunächst die Translationsinvarianz aus und startet somit von jedem Gitterplatz. Dies sorgt für einen Faktor N^4 in der Laufzeit. Für den Loop soll natürlich die maximale Separation der Quarks auf dem Gitter ausgenutzt werden. Dies entspricht im on-axis Fall gerade $N/2$ in zeitlicher und räumlicher Richtung. Da für jede räumliche Separation auch jede zeitliche durchlaufen wird, ergibt sich somit ein Faktor N^2 . Aufgrund der temporalen Eichung, steuern die zeitlichen Linienintegrale für jede Separation den gleichen Rechenaufwand bei. Die räumlichen Linienintegrale bestehen jedoch aus Multiplikationen von Links und haben somit einen Rechenaufwand proportional zu ihrer Länge. Aufgerundet kommt somit ein Laufzeitfaktor N hinzu. Dies macht insgesamt ein Laufzeitverhalten von N^7 .

- **Der modifizierte Wilson Loop**

Für den modifizierten Wilson Loop gilt zunächst das Gleiche, wie für den gewöhnlichen. Auch er nutzt die Translationsinvarianz aus (N^4) und bekommt einen Faktor N^2 durch die räumlichen und zeitlichen Separationen. Da die räumlichen Linienintegrale nun aber durch non-string-like Objekte ersetzt wurden, kommen keine weiteren, von der Gittergröße abhängigen, Faktoren zum Laufzeitverhalten hinzu. Insgesamt ergibt sich somit ein Laufzeitverhalten von N^6 , also eine Potenz niedriger als beim gewöhnlichen Wilson Loop.

- **Der off-axis Fall**

Auch im off-axis Fall bleibt der Faktor N^4 wegen der Translationsinvarianz bestehen. Für den Loop stehen nun aber zwei weitere Dimensionen zur Verfügung. Für die verschiedenen Quark-Separationen ergibt sich daher ebenfalls ein Faktor N^4 . Auch im off-axis Fall wird der modifizierte Wilson Loop verwendet, daher

kommen keine weiteren, von der Gittergröße abhängigen, Faktoren hinzu. Das Laufzeitverhalten beläuft sich also auf N^8 .

Nun haben wir alle relevanten Laufzeiten abgeschätzt. In Abschnitt (4.4) wird sich zeigen, ob diese auch in der Praxis beobachtet werden.

4 Vorstellung der Ergebnisse

In diesem Abschnitt sollen nun einige Ergebnisse der in Kapitel (3) vorgestellten Verfahren präsentiert werden. Für alle in diesem Abschnitt gezeigten Ergebnisse wurde ein periodisches Gitter mit einer räumlichen und zeitlichen Ausdehnung von jeweils 16 Gitterpunkten verwendet. Der Parameter β , welcher den Gitterabstand festlegt, beträgt $\beta = 2.5$. Dies entspricht in etwa einem Gitterabstand von $0.073 fm$ [4]. Es wurden jeweils 100 Feldkonfigurationen verwendet, welche mittels Monte-Carlo-Heatbath-Algorithmus erstellt wurden.

4.1 Ergebnisse unter Verwendung von string-like trial states

Wie in Kapitel (3.1) bereits angesprochen, ist die für uns relevante Größe die effektive Masse. Diese ist in Abbildung (6) für verschiedene Quark-Separationen aufgetragen. Wie man sieht, erfüllt sie genau das Verhalten, welches man aus Gleichung (3.11) vorhersagen kann. Für kleine Zeiten ist noch Variation vorhanden, was dem exponentiellen Abfall höherer Energiemoden entspricht. Für größere Zeiten nähert sich die effektive Masse dann jedoch einem Plateau an.

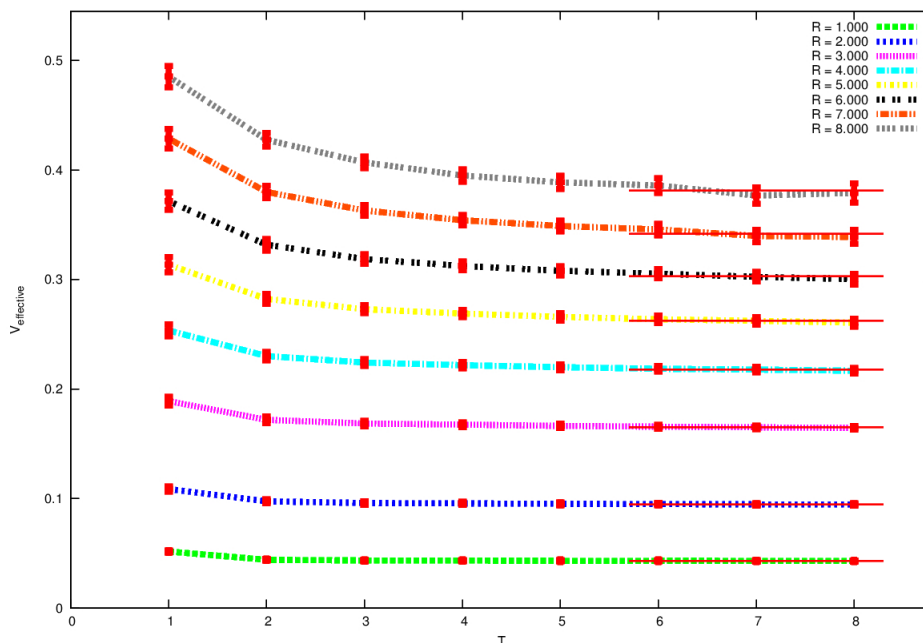


Abbildung 6: Effektive Masse

Es fällt auf, dass die Plateaus für zunehmende Quark-Separationen schlechter werden. Dies lässt sich dadurch erklären, dass die bei großen Separationen konstruierten trial-states einen größeren Überlapp zu höher angeregten Energiemoden besitzen. Somit ist mehr Zeit nötig, bis der niedrigste Energieeigenzustand dominiert und das Plateau sichtbar wird.

Extrahiert man nun aus den effektiven Massen die entsprechenden Energien und trägt diese gegen den Abstand auf, erhält man Abbildung (7). Die Datenpunkte wurden zusätzlich mit der erwarteten Form des Potentials gefittet. Also einer Funktion der Form:

$$V(R) = V_0 - \frac{\alpha}{R} + \sigma \cdot R \quad (4.1)$$

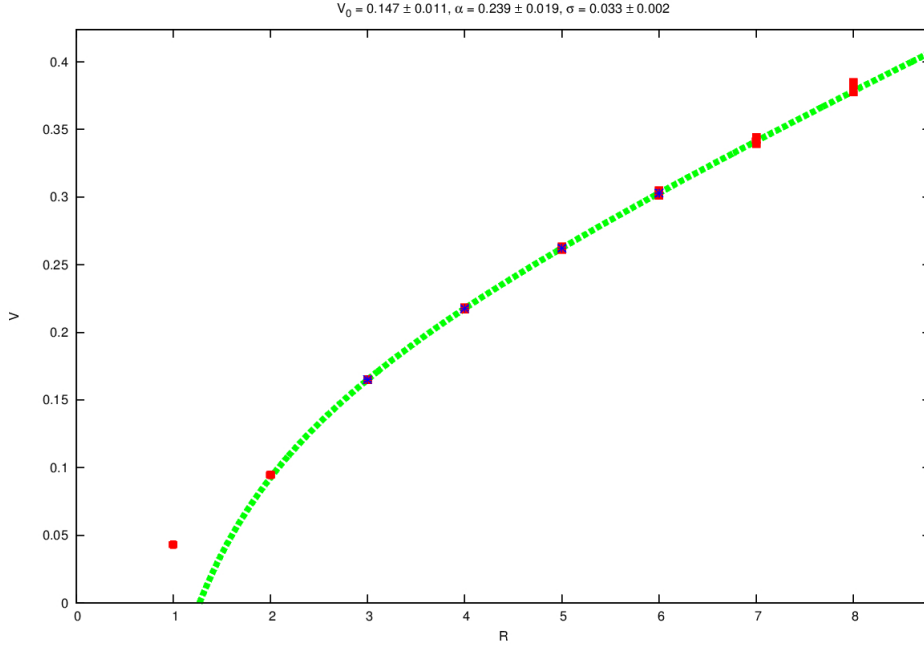


Abbildung 7: Das statische $\bar{Q}Q$ -Potential

Wie man sieht, stimmt dieser Fit sehr gut mit den Datenpunkten überein. Etwas auffällig sind jedoch die Datenpunkte bei kleineren Separationen, welche auch nicht zum Fit beitragen. Dies ist ein bekanntes Problem und auf die Diskretisierung der Raumzeit zurückzuführen. Wie in Kapitel (2.5) bereits erwähnt, muss man sich bewusst sein, dass durch die Diskretisierung so genannte Diskretisierungsfehler entstehen. Diese äußern sich vor allem bei kleinen Abständen und sorgen so für Sprünge im Potentialverlauf. Das gleiche Verhalten werden wir auch in den folgenden Plots, insbesondere bei off-axis Rechnungen, beobachten.

Nun steht noch aus die Qualität der gezeigten Ergebnisse zu überprüfen. Hierzu versuchen wir aus den gefitteten Parametern α und σ auf den verwendeten Gitterabstand zurückzuschließen. Sollte dieser sich in guter Näherung reproduzieren, ist dies ein Anzeichen für ein gutes Ergebnis. Zur Erinnerung, der verwendete Gitterabstand entspricht in etwa $a = 0,073 fm$.

Für die Durchführung benötigen wir den *Sommer-Parameter* $r_0 = 0,46 fm$. Dieser Abstand erfüllt folgende experimentell gemessene Gleichung:

$$\left. \frac{\partial V(R)}{\partial R} \right|_{r_0} \cdot r_0^2 = 1,65 \quad (4.2)$$

In diese Gleichung kann nun der durch Gleichung (4.1) beschriebene Potentialverlauf eingesetzt werden. Die entstehende Gleichung ist daraufhin nach r_0 umzustellen.

$$r_0 = \sqrt{\frac{1,65 - \alpha}{\sigma}} \quad (4.3)$$

Natürlich muss nun beachtet werden, dass wir uns auf dem Gitter befinden. Setzt man also in Gleichung (4.3) die aus Abbildung (7) abgelesenen Potentialwerte ein, erhält man den Sommer-Parameter in Einheiten des Gitterabstandes r_0^G . Dies ermöglicht dann aber auch die Bestimmung des Gitterabstandes durch:

$$a = \frac{r_0}{r_0^G} = \frac{0,46 fm}{r_0^G} \quad (4.4)$$

Für die gefundenen Parameter ergibt sich: $r_0^G = 6,57 \pm 0,13$. Nach Gleichung (4.4) entspricht dies einem Gitterabstand von $a = (0,070 \pm 0,001) fm$. Der so gefundene Gitterabstand besitzt also eine Abweichung von $\Delta a = 4\%$ vom tatsächlich verwendeten Gitterabstand. Dies deutet auf eine gute Qualität des berechneten Potentials hin.

4.2 Ergebnisse unter Verwendung von non-string-like trial states

Auch für das neu entwickelte Verfahren, ist für die Auswertung die effektive Masse zu berechnen. Der entsprechende Plot findet sich in Abbildung (8).

Wie man sieht, ist das Verhalten sehr ähnlich zu dem aus Abbildung (6). Es fällt jedoch auf, dass die Plateauqualität etwas schlechter ist. Die naheliegendste Erklärung dafür ist, dass der non-string-like trial state einen größeren Überlapp zu höher angeregten Zuständen besitzt. Dies lässt sich auch schon durch das steilere Abfallen in niedrigen Zeitbereichen vermuten. Im Großen und Ganzen kann die Plateauqualität aber dennoch als gut bewertet werden.

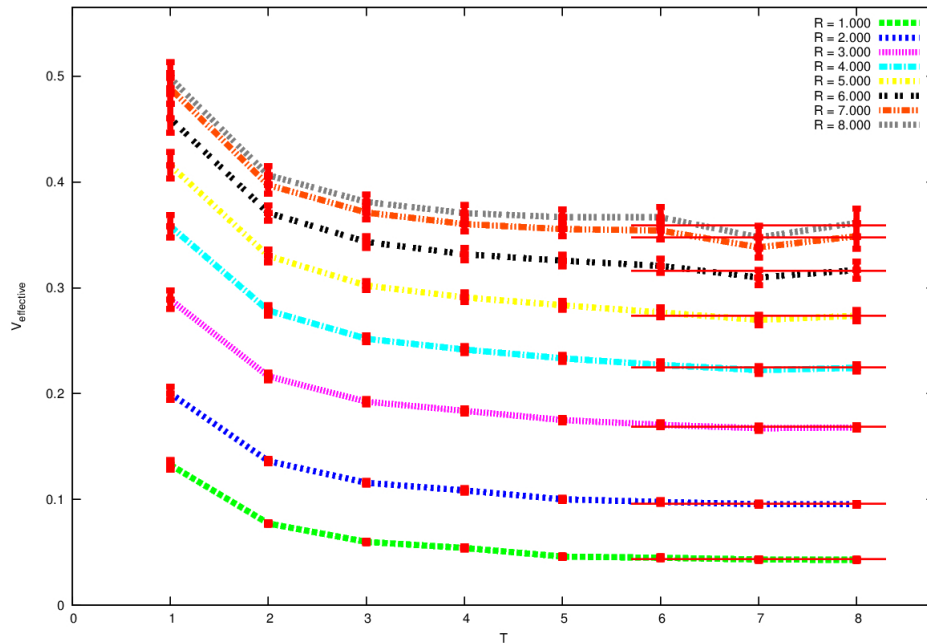


Abbildung 8: Effektive Masse

Aus der effektiven Masse lässt sich nun wieder das Potential extrahieren. Dieses ist in Abbildung (9) dargestellt. Auch hier wurde ein Fit nach Gleichung (4.1) angesetzt. Wie man sehen kann, stimmt auch dieser wieder gut mit den errechneten Datenpunkten überein.

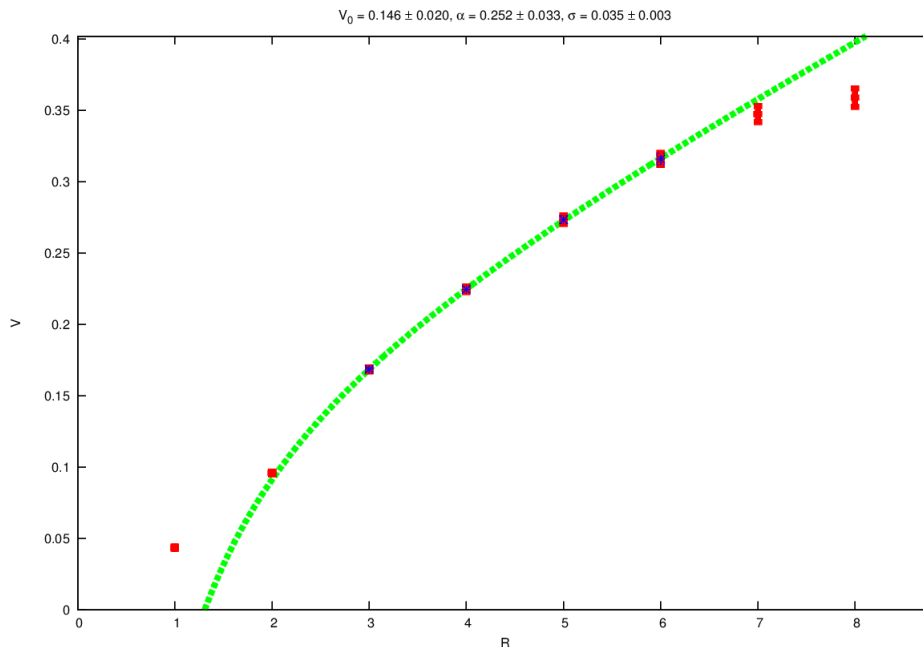


Abbildung 9: Das statische $\bar{Q}Q$ -Potential im neuen Verfahren

Es fällt jedoch auf, dass diesmal auch die Datenpunkte bei großen Abständen Auffälligkeiten aufweisen. Sowohl der Punkt bei $R = 7$ als auch der Punkt bei $R = 8$ ist etwas niedriger, als bei den Ergebnissen aus dem vorherigen Kapitel. Auch dieser Effekt kann jedoch erklärt werden. Grund hierfür ist diesmal die Periodizität des Gitters (finite-size-effects). Für große Separationen tragen so nicht nur die direkten, sondern auch die periodischen

Nachbarn maßgeblich zur Wechselwirkung bei. Effektiv führt das dazu, dass für eine Separation von $R = 8$ keine Kraft wirkt, da sich die links- und rechtsseitigen Kräfte gerade ausgleichen. Dies Resultiert in einem Scheitelpunkt des Potentials, welcher bei dieser Separation beobachtet werden kann.

Natürlich stellt sich die Frage warum dieser Effekt beim Arbeiten mit string-like trial states nicht auftritt, obwohl auch ein periodisches Gitter verwendet wird. Die Antwort hierauf liegt eben gerade im string-like-Charakter des trial states. Durch diesen wird quasi eine Bindungsrichtung vorgegeben, welche die periodischen Effekte zerstört. Insgesamt kann das Verhalten des Potentials aus Abbildung (9) somit erklärt werden.

Auch für das neue Verfahren muss natürlich eine Aussage über die Qualität getroffen werden. Diese kann auf dem gleichen Prinzip wie im vorherigen Abschnitt erfolgen. Die Fit-Parameter können dabei aus Abbildung (9) entnommen werden. Der Gitterabstand ergibt sich dann wieder aus Gleichung (4.4). Man findet in diesem Fall $r_0^G = 6,29 \pm 0,22$. Dies entspricht wiederum einem Gitterabstand von $a = (0,073 \pm 0,003)fm$, welcher hier sogar exakt dem verwendeten Gitterabstand entspricht.

Um noch einen etwas genaueren Blick auf die Unterschiede der Plateauqualität und Fehler der effektiven Masse zu geben, sind diese im folgenden Diagramm nochmal für ausgewählte Abstände zusammen geplottet. Es handelt sich dabei um die Separationen $R = 2$, $R = 4$ und $R = 6$.

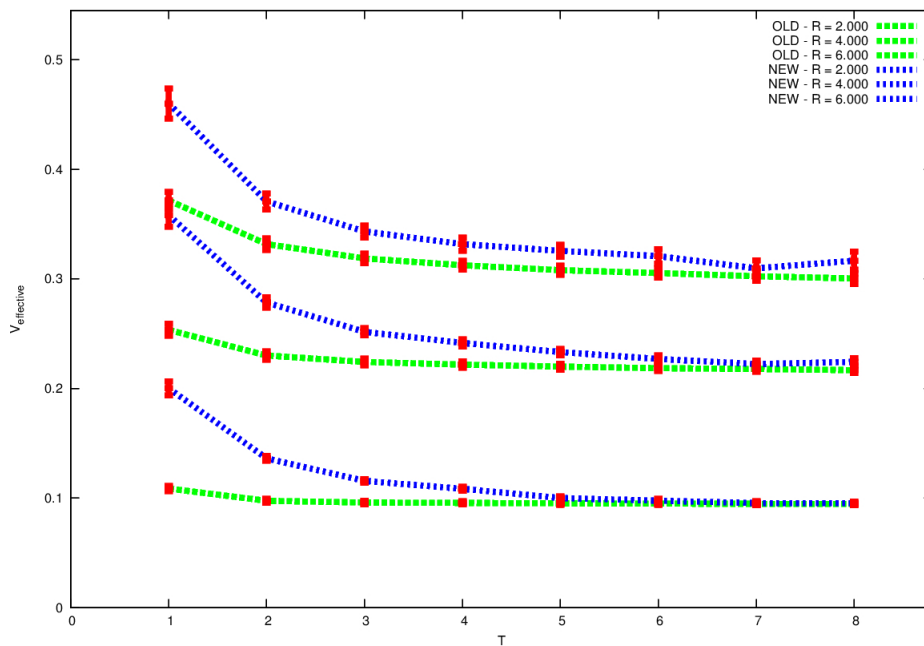


Abbildung 10: Effektive Massen im direkten Vergleich

Die grünen Graphen stellen dabei die effektiven Massen des alten Verfahrens dar, während die Blauen das neue Verfahren repräsentieren. Wie man sieht, ist die Plateauqualität für kleine Separationen annähernd gleich. Das neue Verfahren benötigt lediglich eine Längere Zeit zur Konvergenz, wie weiter oben schon diskutiert wurde. Dieses Verhalten wird zu größeren Separationen immer deutlicher, da dort der Überlapp zu den höher angeregten Energiemoden zunimmt. Weiterhin scheinen auch die statistischen Fehler des neuen Verfahrens, angedeutet durch die Fehlerbalken in Abbildung (10), etwas größer zu sein.

4.3 Ergebnisse für off-axis Abstände

Die bisher gezeigten Ergebnisse besitzen nur eine Auflösung bis hin zu ganzzahligen Gitterabständen. Wie bereits erwähnt ermöglichen off-axis Wilson Loops auch die Berechnung von Zwischenabständen. Auch hier ist die relevante Größe die effektive Masse. Da es diesmal eine Vielzahl von möglichen Separationen gibt, ist der Plott aller effektiven Massen entsprechend unübersichtlich, wie in in Abbildung (11) zu sehen ist. Daher findet sich darunter noch einmal der Plott für ein paar ausgewählte Separationen.

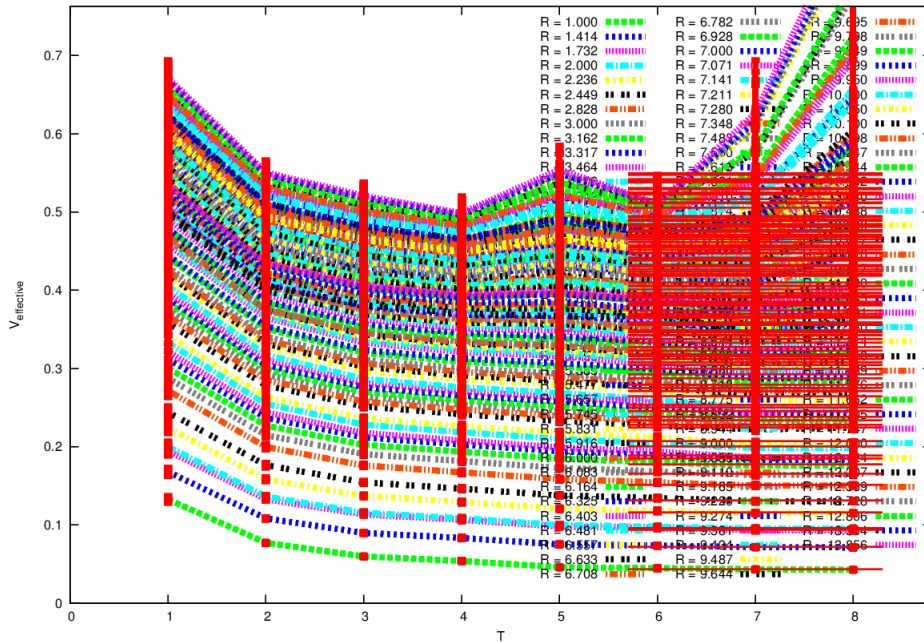


Abbildung 11: Effektive Masse bei off-axis Separationen

Man sieht, dass im Wesentlichen gleiches gilt wie zuvor schon für das neue Verfahren im on-axis Fall gesagt wurde. Für kleine räumliche Separationen ergeben sich recht gute Plateaus, welche mit zunehmender räumlicher Separation schlechter werden.

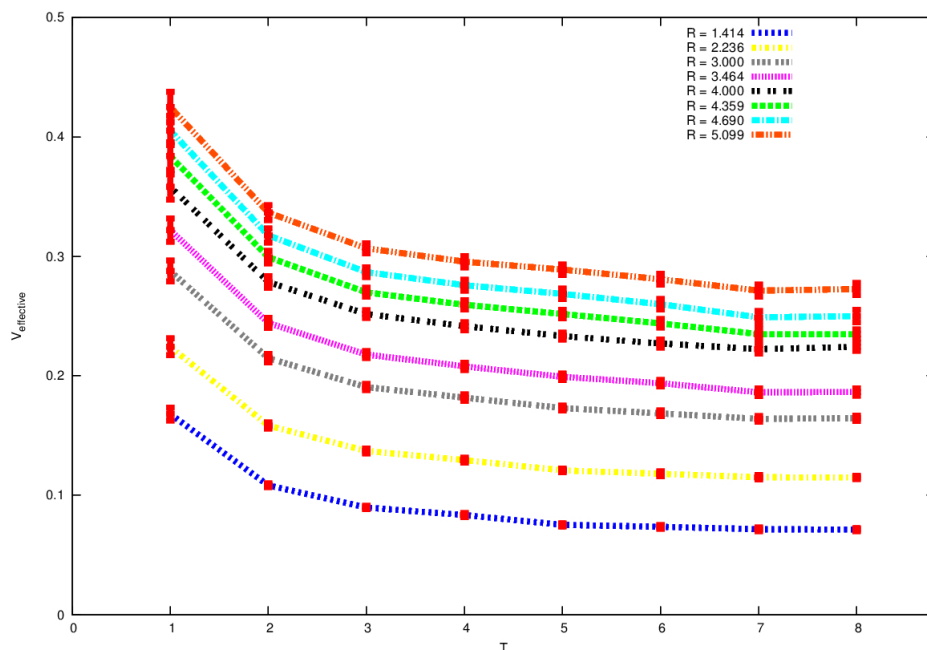


Abbildung 12: Effektive Masse bei off-axis Separationen für ausgewählte Abstände

Auch hier kann das Potential nun wieder einfach über die entsprechenden Plateaus extrahiert werden. Durch die große Menge an Plateaus entsteht so ein sehr feiner Verlauf von Datenpunkten, welcher in Abbildung (13)

dargestellt ist. Es wurde auch wieder ein Fit nach Gleichung (4.1) in den Graphen eingezeichnet. Natürlich muss beachtet werden, dass auch im off-axis Fall Diskretisierungsfehler und finite-size-effects auftreten, welche sich bei kleinen und größeren räumlichen Separationen bemerkbar machen. Daher sollten die in den Fit eingehenden Punkte sorgsam ausgewählt werden.

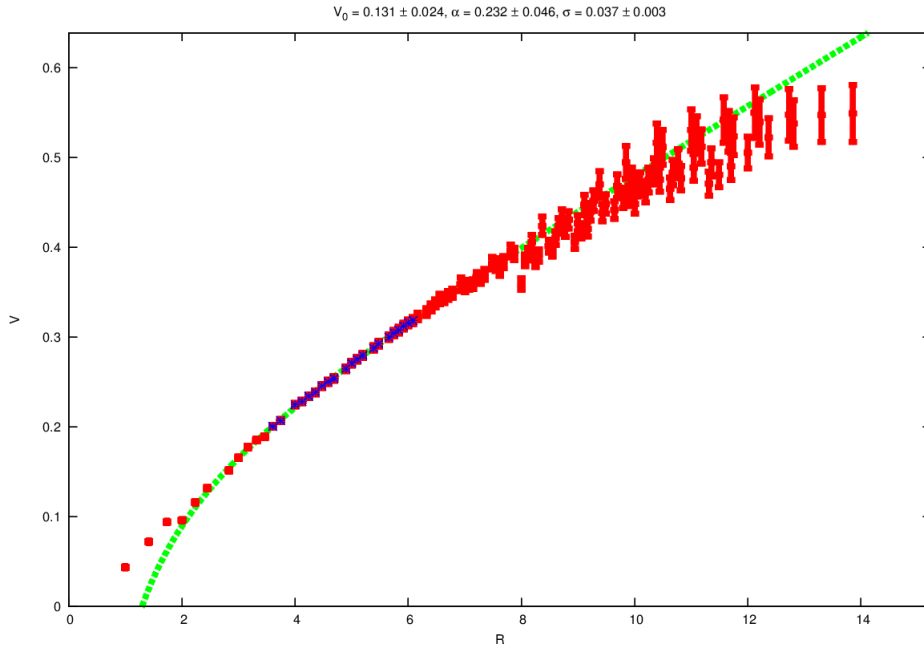


Abbildung 13: Das statische $\bar{Q}Q$ -Potential im off-axis Fall

Auch im off-axis Fall kann natürlich die Frage nach der Qualität der Ergebnisse gestellt werden. Hierzu ist wieder ein Vergleich zwischen verwendeten und errechneten Gitterabstand anzufertigen. Der Gitter-Sommer-Parameter ergibt sich in diesem Fall zu $r_0^G = 6,20 \pm 0,18$. Daraus folgt ein Gitterabstand von $a = (0,074 \pm 0,002) fm$. Dies entspricht einer Abweichung von $\Delta a = 2,95\%$ und deutet somit wieder auf eine gute Potentialqualität hin.

4.4 Laufzeitverhalten

Neben dem extrahierten Potential ist eine weitere relevante Größe das Laufzeitverhalten. Um dieses zu erfassen, müssen die oben gezeigten Rechnungen auf Gitter mit verschiedener Größe übertragen werden. Die Laufzeit kann dann in Abhängigkeit der Gittergröße aufgetragen werden. Dabei gehen wir jeweils von quadratischen Gittern aus, also Gittern, welche die Gleiche räumliche und zeitliche Ausdehnung besitzen. Die so erhaltenen Ergebnisse können dann mit dem theoretischen Laufzeitverhalten aus Abschnitt (3.6) verglichen werden.

- **Eigenvektorrechner:** Für den Eigenvektorrechner konnte das Laufzeitverhalten leider nicht theoretisch vorhergesagt werden. In Abbildung (14) sind die gemessenen Daten für verschiedene Gittergrößen dargestellt. Zusätzlich wurden die Datenpunkte mit einem Polynom vom Grad n gefittet. Der bestmögliche Fit ergab sich für einen Exponenten von $n = 7$. Dieses Ergebnis ist natürlich mit Vorsicht zu behandeln, da die Menge an Datenpunkten für eine gefestigte Aussage nicht ausreichend ist.

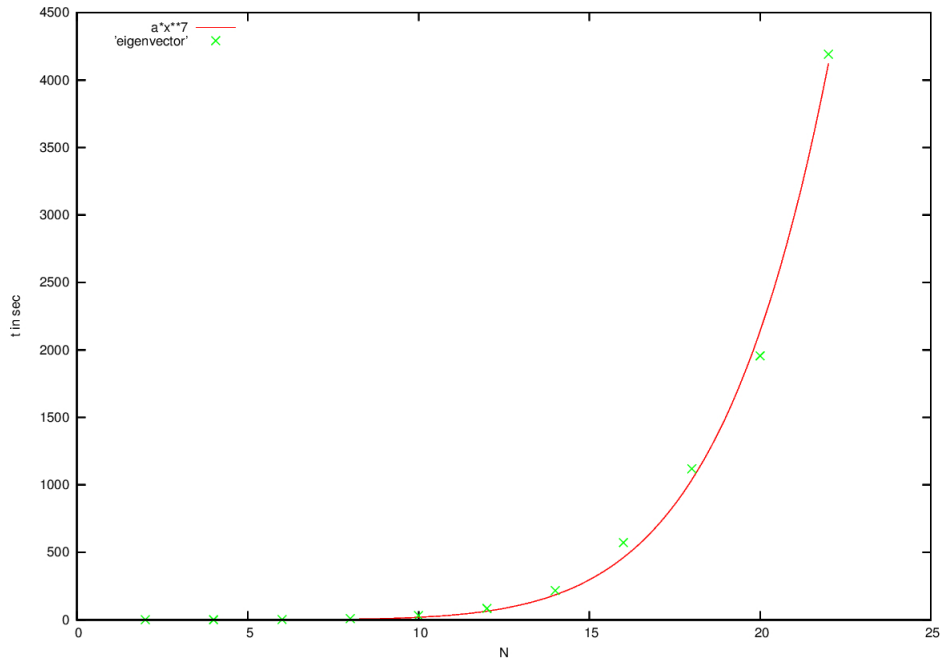


Abbildung 14: Laufzeit des Eigenvektorrechners

- **Der gewöhnliche Wilson Loop:** Für den gewöhnlichen Wilson Loop wurde ein Laufzeitverhalten von N^7 vorhergesagt. Auch hier wurde an die Daten ein Polynom vom Grad n gefittet. Dies führte zu einem Ergebnis von $n = 6,61$, welches recht gut mit der theoretischen Vorhersage übereinstimmt. In Abbildung (15) sind die gemessenen Datenpunkte aufgetragen und mit einem Polynom vom Grad $n = 7$ gefittet.

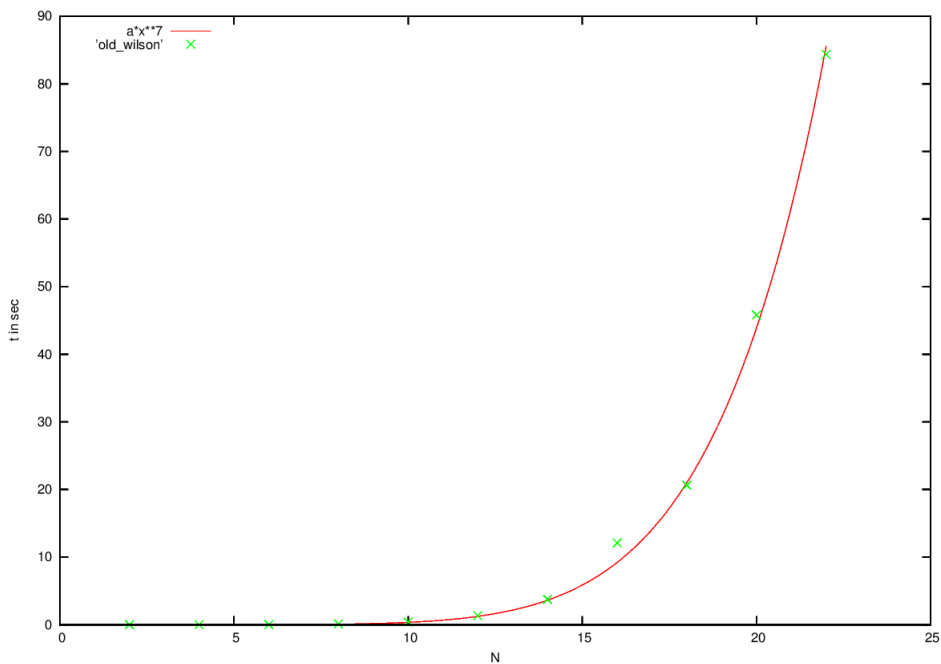


Abbildung 15: Laufzeit der gewöhnlichen Wilson Loop

- **Der modifizierte Wilson Loop:** Für den modifizierten Wilson Loop wurde ein Laufzeitverhalten von N^6 vorhergesagt. Auch hier wurde die Prozedur von oben wiederholt und ein Polynom von Grad n an die Datenpunkte gefittet. Dies ergab einen Exponenten von $n = 5,79$, was ebenfalls konsistent mit der theoretischen Vorhersage ist. In Abbildung (16) sind wieder die Datenpunkte aufgetragen und mit einem Polynom von Grad $n = 6$ gefittet.

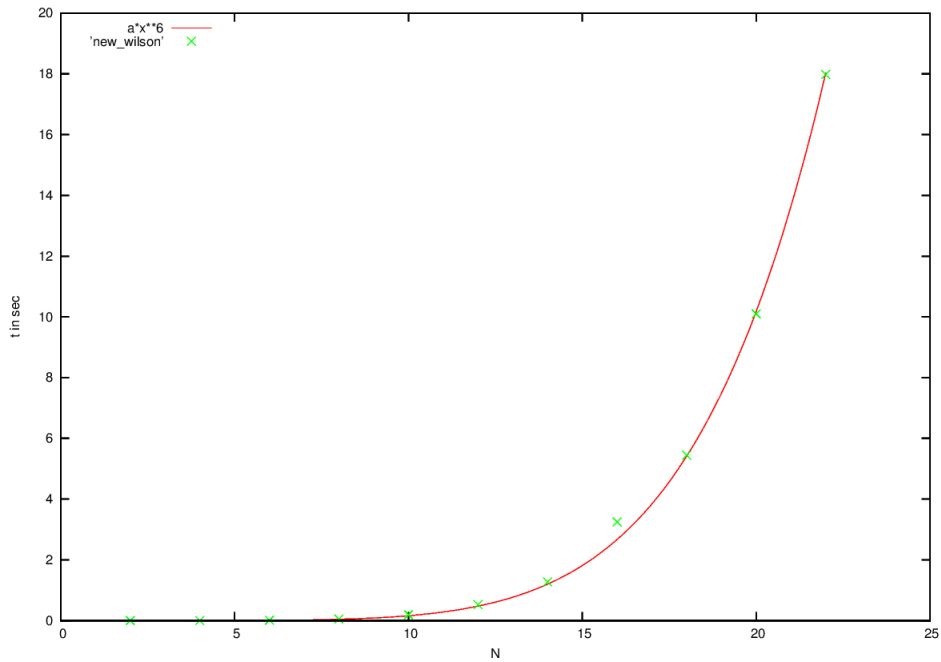


Abbildung 16: Laufzeit der modifizierten Wilson Loop

- **Direkter Vergleich beim on-axis Fall:** Für den on-axis Fall sind Laufzeitmessungen sowohl für das alte Verfahren, als auch für das neue Verfahren vorhanden und wurden oben vorgestellt. Natürlich bietet sich noch ein direkter Vergleich an, welcher beide Laufzeitverhalten in einem Plot darstellt. Dieser findet sich in Abbildung (17). Die grüne Kurve repräsentiert dabei den gewöhnlichen Wilson Loop, während die Rote das Laufzeitverhalten des modifizierten Verfahrens zeigt.

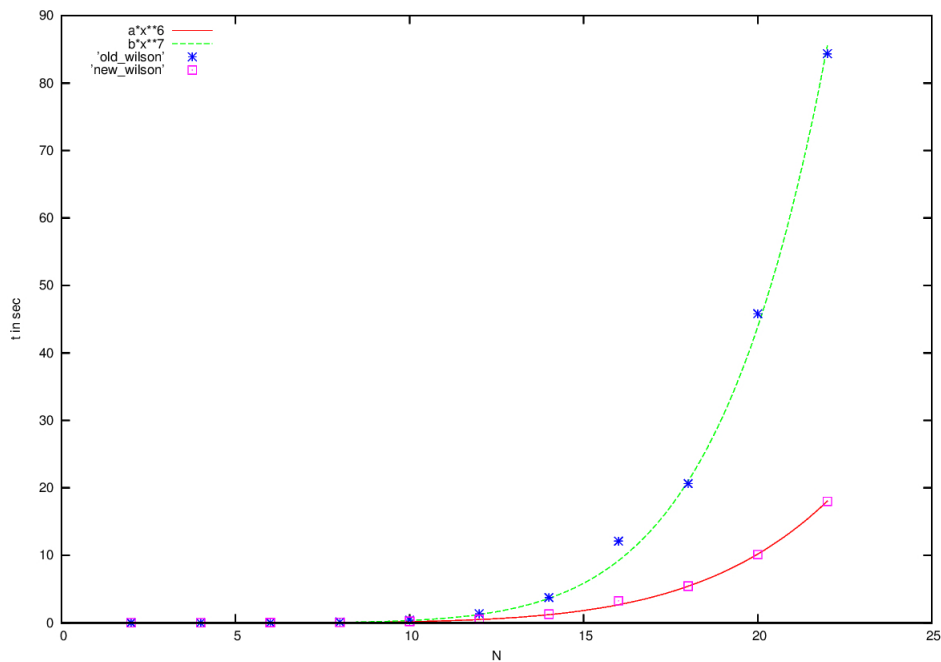


Abbildung 17: Direkter Vergleich des Laufzeitverhaltens

- **Der off-axis Fall:** Für den off-axis Fall wurde das Laufzeitverhalten zu N^8 abgeschätzt. Fittet man auch hier ein Polynom vom Grad n an die Datensätze, bekommt man für den Exponenten einen Wert von $n = 8,08$. Auch hier herrscht also gute Übereinstimmung zu den theoretischen Vorhersagen. In Abbildung (18) findet sich die Laufzeit gegen die Gittergröße aufgetragen, mit einem Polynom Fit vom Grad $n = 8$.

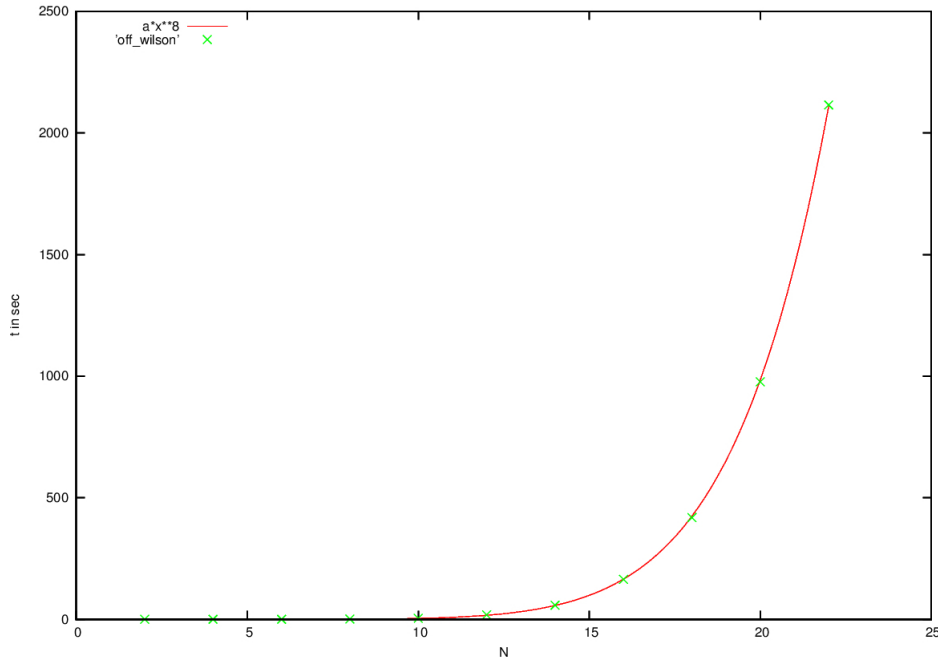


Abbildung 18: Laufzeit für den off-axis Fall

5 Diskussion der Ergebnisse

Im vorherigen Abschnitt wurden nun einige Ergebnisse, der aus Abschnitt (3) vorgestellten Verfahren, präsentiert. Für die Diskussion ist insbesondere der direkte Vergleich zwischen dem alten Verfahren aus (3.1) und dem neuen Verfahren aus (3.2), welches die neu eingeführten non-string-like trial states verwendet, von Interesse.

Wie wir gesehen haben, produzieren beide Verfahren ähnliche Ergebnisse. Sowohl die effektiven Massen, als auch die Potentialverläufe verhalten sich wie erwartet und liefern beim Zurückführen auf den Gitterabstand gute Resultate. Im direkten Vergleich von altem und neuem Verfahren zeigen sich jedoch ein paar Schwachstellen der neuen Methode. Wie wir in Abbildung (10) gesehen haben, ist die Plateauqualität nicht so gut wie beim alten Verfahren und auch die statistischen Fehler erweisen sich als größer. Allerdings muss man auch dazu sagen, dass die neue Methode, aufgrund des höheren Überlapps zu angeregten Zuständen, das Plateau erst bei späteren Zeiten erreicht. Dies könnte dazu führen, dass die beiden Verfahren auf zeitlich weiter ausgedehnten Gittern, zu einer besseren Übereinstimmung kommen.

Für off-axis Abstände konnte das neue Verfahren vor allem durch seinen anschaulichen Zugang und die einfache Implementierung überzeugen. Auch die in Kapitel (4.3) enthaltenen Ergebnisse weisen die erwartete Form auf und lieferten, im Test mit dem Sommer-Parameter, ein gutes Ergebnis.

Was für die Diskussion noch ausbleibt, ist eine Betrachtung des Laufzeitverhaltens. Dies war schließlich mit ein Grund für die Entwicklung des neuen Verfahrens. Wie wir in Kapitel (4.4) gesehen haben, konnte das Laufzeitverhalten für den Wilson Loop tatsächlich um einen Faktor N (Gittergröße) reduziert werden. Besonders auf großen Gittern ergibt sich so eine immense Zeitersparnis, wie in Abbildung (17) zu erkennen ist. Dies allein ist allerdings nur die halbe Wahrheit, denn für die Berechnung mit dem neuen Verfahren werden natürlich noch die Eigenvektoren des kovarianten Laplaceoperators benötigt. Deren Berechnung nimmt eine relativ große Zeit in Anspruch und das Laufzeitverhalten ist nicht genau bestimmt.

Somit ist der Vorteil des neuen Verfahrens eher in der off-axis Rechnung zu suchen, da hier der Rechenaufwand der Loops größer wird, während der für die Eigenvektoren gleich bleibt. Hier ergibt sich jedoch das Problem, dass kein direkter Vergleich vorliegt. Betrachtet man allerdings Ergebnisse anderer Arbeiten, welche off-axis Rechnungen mit gewöhnlichen Wilson Loops verwenden z.B. [5], ergibt sich ein sehr deutlicher Laufzeitunterschied zu Gunsten des neuen Verfahrens. Natürlich werden solche Vergleiche aber erst dann aussagekräftig, wenn die gleichen Rechner verwendet werden und die Implementierung des Codes auf beiden Seiten optimiert wurde. Demnach ist eine endgültige Aussage über das Laufzeitverhalten nicht möglich.

Zusammengefasst lässt sich also sagen, dass das neue Verfahren durchaus vielversprechende Tendenzen besitzt. Es überzeugt durch seine anschauliche und einfache Implementierung, liefert vergleichbar gute Ergebnisse und könnte hinsichtlich der Laufzeit eine Zeitersparnis einbringen.

6 Programm Dokumentation

Im folgenden Abschnitt soll ein Überblick der verwendeten Programme gegeben werden. Für die richtige Funktionsweise ist es notwendig, dass auf dem verwendeten Rechner das ARPACK (Arnoldi-Package) zur Verfügung steht. Näheres dazu kann in [3] nachgelesen werden. Weiterhin müssen die in den Programmen enthaltenen Dateipfade auf den verwendeten Rechner angepasst werden.

6.1 Grundlegende-Programme

In diesem Kapitel werden Programme vorgestellt, die im gesamten Code Anwendung finden. Sie dienen zur Speicherverwaltung, Ein- und Ausgabe und enthalten elementare Rechenoperationen. Viele Funktionen wurden dabei von einem schon bestehenden Code übernommen und für diese Arbeit entsprechend vervollständigt.

- **io.cc**
Der Code, welcher in io.cc (in- and output) enthalten ist, übernimmt das Einlesen und Ausschreiben von Eichfeldkonfigurationen und Eigenvektoren.
- **fields.cc**
fields.cc übernimmt die dynamische Speicherreservierung für Feldkonfigurationen und Eigenvektoren. Weiterhin enthält es Funktionen, die eine Eichtransformation der Feldkonfiguration durchführen können, insbesondere auch eine für temporale Eichung.
- **linear_algebra.hh**
In dieser Bibliothek sind alle nötigen Rechenoperationen enthalten. Dies umfasst sowohl Operationen im Raum der Eichfelder, als auch auf dem Raum der Eigenvektoren. Insbesondere enthält diese Bibliothek auch den Aufruf für die Berechnung der Eigenvektoren.
- **geometry.hh**
Diese Bibliothek ist für die Speicherverwaltung zuständig. Sowohl Feldkonfigurationen als auch Eigenvektoren werden als ein Array gespeichert. Um durch Angabe eines Gitterpunktes zum entsprechenden Speicherelement springen zu können, sind die Funktionen aus geometry.hh notwendig.

6.2 Smearing

Für das neue Verfahren ist es sehr wichtig, dass das Smearing vor der Eigenvektorberechnung erfolgt. Durch die Änderung der räumlichen Links, aufgrund des APE-smearing würden sonst die Eigenvektoren verloren gehen. Um die temporale Eichung zu erhalten, muss das HYP-smearing ebenfalls im Voraus erfolgen, da ein nachträgliches HYP-smearing die temporale Eichung zerstören würde.

- **HYP_smearing.C**
Führt ein HYP-smearing der übergebenen Konfigurationen durch.
- **APE_smearing.C**
Führt ein APE-smearing der übergebenen Konfigurationen durch.

6.3 Berechnung der Eigenvektoren

Die Berechnung der Eigenvektoren wurde hauptsächlich von einem anderen Code übernommen [3]. Dennoch gibt es in dem verwendeten Code einige nennenswerte Änderungen.

- **eigenvectorcalc.cpp**
Dies ist das Programm, welches zur Berechnung der Eigenvektoren aufgerufen werden muss. Es ruft dann entsprechend die weiter unten genannten Programme auf, um die Eigenvektoren zu berechnen.
- **eigenvector.hh, znaupd.h**
Dies sind die übernommenen Programme in denen die eigentliche Berechnung der Eigenvektoren stattfindet. Näheres zu diesen Programmen kann man in [3] nachlesen.

- **conversation.cc**

Die von `eigenvector.hh` berechneten Eigenvektoren sind vom C++-Typ `'complex< double >'`. Da dies zu Komplikationen mit dem bisher vorgestellten Code führt, übernimmt `conversation.cc` das Umschreiben auf ein `double` Array. Näheres dazu in Kapitel (6.6).

- **proof.cc**

`proof.cc` übernimmt einen entscheidenden Teil bei der Berechnung der Eigenvektoren. Es prüft noch einmal, ob die ausgegebenen Eigenvektoren auch wirklich Eigenvektoren sind. Dieser Test ist sowohl hinter die Eigenvektorberechnung, als auch hinter jeden Einlesevorgang von Eigenvektoren geschaltet.

6.4 Berechnung des Wilson Loop

Nun kommt es zum Kernstück dieser Arbeit. Die Berechnung des Wilson Loop mithilfe der Eigenvektoren des kovarianten Laplaceoperators. Dies wird durch die folgenden Programme realisiert.

- **wilson_loop_new.cpp**

In diesem Programm findet die Berechnung des Wilson Loop statt. Ob dieser mit der neuen oder alten Methode ausgerechnet werden soll, kann über das Define-Kommando ausgewählt werden. Die jeweilige Berechnung folgt dann genau wie im Abschnitt (3.2) beschrieben.

- **wilson_loop_averages.cpp**

Wie der Name schon andeutet, werden hier die Mittelwerte des Wilson Loop über eine Konfiguration berechnet. Dazu wird der Wilson Loop in jedem Punkt des Gitters und über jede gewünschte Länge ausgerechnet und danach der Mittelwert gebildet. Die Ergebnisse werden in einer Datei gespeichert.

- **Off-Axis**

Die Beiden oben genannten Programme stehen auch als off-axis Version zur Verfügung. Diese Varianten ermöglichen die Berechnung von nicht ganzzahligen Gitterabständen.

6.5 Programmtests

Um Fehler systematisch auszuschließen, ist es wichtig die Funktionsweise der Programme an mehreren Stellen zu überprüfen. Der Theorieteil in Abschnitt (3) liefert einige Möglichkeiten zur Überprüfung, welche durch folgende Codes durchgeführt werden.

- **gauge_transform.cpp**

Dieser Code nutzt die schon angesprochene Funktion aus `linear_algebra.hh` und führt eine Eichtransformation von einer Feldkonfiguration durch. Sowohl das auf diese Weise entstandene Feld, als auch die verwendete Transformation werden in einer Datei gespeichert. Dies ist für die Überprüfung der Ergebnisse essenziell, denn damit kann die Eichinvarianz des Wilson Loop oder das Transformationsverhalten der Eigenvektoren geprüft werden.

- **vector_test.cpp**

In diesem Programm werden mehrere Programmtests zusammengefasst. Dies ist die Überprüfung auf den richtigen Entartungsgrad der Eigenwerte, sowie der Test auf Orthonormalität und das Transformationsverhalten der Eigenvektoren.

6.6 Speicherverwaltung

Um den Programmcode zu verstehen, ist es zunächst notwendig die Speicherverwaltung zu verstehen. Daher soll diese kurz erklärt werden. Die Eichfeldkonfigurationen werden als `'double *'` gespeichert, also als einfaches `double` Array. Die einzelnen Werte können also durch anfügen eines Index aufgerufen werden. Wie dieser Index zu wählen ist, wird durch `geometry.hh` erklärt. Die in dieser Bibliothek vorhandenen Funktionen liefern durch Angabe der Koordinaten eines Punktes den entsprechenden Index.

Die Speicherverwaltung der Eigenvektoren ist etwas komplizierter gehandhabt. Ein Eigenvektor wird als `'double ***'` gespeichert. Es sind daher drei Indizes anzuhängen, um bei einem Wert zu landen. Der erste Index beschreibt die Zeitkomponente zu welcher der Eigenvektor gehört. Der zweite Index steht für die Auswahl der entarteten Eigenvektoren. Der Dritte ist letztendlich wieder für die räumlichen Koordinaten zuständig und kann wieder über die in `geometry.hh` enthaltenen Funktionen angesprochen werden.

Dabei ist zu beachten, dass die Eigenvektoren an jedem Raumpunkt einen zweikomponentigen Vektor im Farbraum bilden. Somit verbraucht jeder Raumpunkt vier Speicherelemente. Es liegen Realteil und Imaginärteil einer Farbkomponente hintereinander, gefolgt von der anderen Farbkomponente.

Die Eigenwerte sind ähnlich verwaltet. Sie werden als `'double **'` gespeichert, benötigen also zwei Indizes. Auch

hier steht der erste Index für die Zeitkomponente und der Zweite für die Auswahl der entarteten Eigenwerte. Dabei liegen Real- und Imaginärteil eines Eigenwertes hintereinander im Speicher, sodass bei der Wahl des zweiten Index Vorsicht geboten ist.

6.7 Verwendung des Programms

Zu guter Letzt soll nun noch eine kleine Anleitung für die verwendeten Programme gegeben werden. Vorausgesetzt wird das Vorhandensein einer hinreichend großen Anzahl von Feldkonfigurationen (MC-Heatbath-Algorithmus). Gestartet wird dann mit dem Smearing der Konfigurationen, gefolgt von der Eichtransformation in temporale Eichung. Danach kann `eigenvectorcalc.cpp` verwendet werden, um die Eigenvektoren zu bestimmen. Nachdem die Eigenvektoren verfügbar sind, kann auch schon das Programm `wilson_loop_averages.cpp` aufgerufen werden. Dieses berechnet über jede Feldkonfiguration den Mittelwert des Wilson Loop unter Verwendung der zugehörigen Eigenvektoren. Die jeweiligen Mittelwerte werden in Dateien gespeichert. Um zu den oben gezeigten Graphen zu gelangen, ist es dann noch notwendig den Mittelwert über alle vorhandenen Konfigurationen zu bilden und die effektiven Massen zu bestimmen. Durch einen Plateau-Fit kann dann das Potential extrahiert werden.

Literatur

- [1] Michael E. Peskin and Dan V. Schroeder. *An Introduction To Quantum Field Theory (Frontiers in Physics)*. Westview Press, 1995.
- [2] Heinz J. Rothe. *Lattice Gauge Theory - An Introduction*. World Scientific, 2012.
- [3] Janik Kämper. Non-string-like trial states for static potential computations. Bachelorarbeit, Goethe Universität Frankfurt, 2014.
- [4] Marc Wagner Owe Philipsen. On the definition and interpretation of a static quark anti-quark potential in the colour-adjoint channel. *Phys. Rev. D* 89, 014509 (2014), 2013.
- [5] Sumir Motreedja. Numerische bestimmung des statischen potentials für nicht-axiale quark-antiquark-separationen. Bachelorarbeit, Goethe Universität Frankfurt, 2014.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt wurde.

Ort, Datum

Unterschrift