



Institut für Theoretische Physik

**Numerische Bestimmung des statischen Potentials
für nicht-axiale Quark-Antiquark Separationen**

Bachelorarbeit

eingereicht von:
Sumir Motreedja

Dezember 2014

Betreuer & Erstgutachter: Prof. Dr. Marc Wagner
Zweitgutachter: Prof. Dr. Owe Philipsen

Die Zielsetzung der vorliegenden Arbeit ist die numerische Bestimmung des statischen Potentials für nicht-ganzzahlige Quark-Antiquark Abstände im Rahmen der $SU(2)$ -Yang-Mills-Theorie. Die Arbeit führt vor allem in die Theorie der nicht-axialen Wilson Loops und in die numerische Bestimmung von Erwartungswerten ein. Ein weiterer Teil dieser Arbeit beschäftigt sich mit der Effizienz und Erweiterbarkeit der numerischen Potentialbestimmung auf die $SU(3)$ -QCD.

Inhaltsverzeichnis

1. Einführung in die Theorie: Die Bestimmung des statischen Quark-Antiquark Potentials

1.0 Der Wilson Loop und das statische Potential.....	1-2
1.1 Der on-axis Wilson Loop.....	3-4
1.2 Der off-axis Wilson Loop.....	4-5
1.3 Die numerische Bestimmung des Erwartungswerts des Wilson Loops.....	5-6

2. Numerisches Konzept: Die Implementierung des off-axis Wilson Loops und die Effizienz

2.0 Erklärungen zur Implementierung des off-axis Wilson Loops.....	6-11
2.1 Betrachtung von effizienzsteigernden Methoden.....	12-13
2.2 Weitere Methoden zur Effizienzsteigerung.....	14

3. Ergebnisse und Diskussion: Das statische Quark-Antiquark Potential für nicht-axiale Separationen

3.0 Beschreibung der Graphen zum statischen Potential.....	15-18
3.1 Qualitative Betrachtung der Ergebnisse.....	18-19
3.2 Abschließende Gesamtaussage.....	19

Literaturverzeichnis.....	20
---------------------------	----

Hinweise zum Umgang mit dem Programmpaket.....	20
------------------------------------------------	----

1. Einführung in die Theorie: Die Bestimmung des statischen Quark-Antiquark Potentials

1.0 Der Wilson Loop und das statische Potential [1]

Die Betrachtung des statischen Quark-Antiquark Potentials erfolgt hier im Rahmen der Yang-Mills-Theorie. Es handelt sich dabei um eine nicht-abelsche Feldtheorie, die invariant ist unter lokalen Transformationen der Gruppe SU(2) [2]. Um im Allgemeinen Observablen, wie Hadronmassen, Glueballmassen oder in diesem Fall das statische Quark-Antiquark Potential zu bestimmen, muss ein passender Operator in der Korrelationsfunktion verwendet werden. Zur Bestimmung des statischen Quark-Antiquark Potentials betrachte man die folgende Korrelationsfunktion:

$$\langle \Omega | \hat{O}^+(t) \hat{O}(0) | \Omega \rangle \text{ mit dem Operator } \hat{O}(t) = \bar{Q}(\vec{x}, t) U(\vec{x}, t; \vec{y}, t) Q(\vec{y}, t) ,$$

wobei $|\Omega\rangle$ der Grund- bzw. Vakuumzustand des quantenmechanischen Systems, Q ein Quarkoperator, \bar{Q} ein Antiquarkoperator und $U(\vec{x}, t; \vec{y}, t) = P \exp\left(ig \int_{\vec{x}}^{\vec{y}} dz^i A_i(\vec{z}, t)\right)$ eine Linkvariable ist (mit matrixwertigem SU(2)-Eichfeld A_i und Pfadordnungsoperator P). Nun wird die Zeitabhängigkeit des Operators $\hat{O}^+(t)$ mit Zeitentwicklungsoperatoren ausgedrückt, dann wird ein vollständiger Satz von Energieeigenzuständen eingeschoben:

$$\langle \Omega | \hat{O}^+(t) \hat{O}(0) | \Omega \rangle = \sum_n \langle \Omega | e^{+Ht} \hat{O}^+(0) e^{-Ht} | n \rangle \langle n | \hat{O}(0) | \Omega \rangle = \sum_n \left| \langle \Omega | \hat{O}^+(0) | n \rangle \right|^2 \exp(-(E_n - E_\Omega)t)$$

Für große euklidische Zeiten t werden höhere Energiezustände durch die Exponentialfunktion unterdrückt, es dominiert der Term im Grundzustand $|0\rangle$, d.h.:

$$\lim_{t \rightarrow \infty} \langle \Omega | \hat{O}^+(t) \hat{O}(0) | \Omega \rangle = \left| \langle \Omega | \hat{O}^+(0) | 0 \rangle \right|^2 \exp(-(E_0 - E_\Omega)t) \quad (\text{Glg. 1.0}),$$

wobei $E_0 - E_\Omega = V_{Q\bar{Q}}(r)$, $r = |\vec{x} - \vec{y}|$. Im Limes großer Zeiten lässt sich somit aus dem exponentiellen Abfall das statische Quark-Antiquark Potential $V_{Q\bar{Q}}(r)$ extrahieren. Vorher jedoch muss die Korrelationsfunktion berechnet werden, um das Potential bestimmen zu können.

Die Korrelationsfunktion sieht in der Pfadintegraldarstellung wie folgt aus:

$$\langle \Omega | \hat{O}^+(t) \hat{O}(0) | \Omega \rangle = \frac{1}{Z} \int D A D Q D \bar{Q} (\bar{Q}(\vec{y}', t) U(\vec{y}', t; \vec{x}', t) Q(\vec{x}', t) \bar{Q}(\vec{x}, 0) U(\vec{x}, 0; \vec{y}, 0) Q(\vec{y}, 0)) e^{iS} ,$$

wobei die Normierungskonstante Z dem obigen Integralausdruck ohne den Klammerterm entspricht und S die Wirkung der schweren Quarkfelder und des SU(2)-Eichfeldes beschreibt. Weiterhin ist eine Integration über die schweren Quarkfelder und über das Eichfeld durchzuführen. Führt man die Integration über die Quarkfelder durch, wendet den statischen Limes (Quarkmasse $M_Q \rightarrow \infty$) an und geht in die euklidische Zeit über, so erhält man das folgende Result:

$$\begin{aligned} & \langle \Omega | \hat{O}^+(t) \hat{O}(0) | \Omega \rangle \propto \langle W_C[A] \rangle_{eukl} \\ & = \frac{\int DA W_C[A] \exp(-S_{YM}^{eukl}[A])}{\int DA \exp(-S_{YM}^{eukl}[A])} \quad (\text{Glg. 1.1}), \end{aligned}$$

wobei S der Yang-Mills-Wirkung entspricht und:

$$W_C[A] = P \exp(ig \oint dz_\mu A_\mu(z)) \quad (\text{Glg. 1.2})$$

ist. Das Wegintegral in $W_C[A]$ wird entlang eines rechteckigen Pfades C ausgeführt. Dies ist der Wilson Loop in der Kontinuumsformulierung. Zur numerischen Bestimmung des statischen Potentials muss die Gitterversion der Glg. 1.1 und 1.2 betrachtet werden. Dabei gilt:

$$W_C[U] = W(\hat{R}, \hat{T}) = \frac{1}{2} \text{Tr} \left(\prod_{l \in C} U_l \right) \quad (\text{Glg. 1.3})$$

Dies entspricht dem Wilson Loop auf dem Raumzeitgitter. Es handelt sich bei den Linkvariablen U_l um Matrizen der Gruppe SU(2). Diese Gruppenelemente vertauschen nicht, daher müssen die Matrizen entlang des Pfades multipliziert werden. Die Spurbildung garantiert, dass der Wilson Loop eine eichinvariante Größe ist. Der Erwartungswert dieser Größe im Grundzustand bzw. die Gitterversion von Glg. 1.1 sieht wie folgt aus:

$$\langle W_C[U] \rangle_{eukl} = \langle W(\hat{R}, \hat{T}) \rangle = \frac{\int DU W_C[U] \exp(-S_{YM}^{eukl}[U])}{\int DU \exp(-S_{YM}^{eukl}[U])} \quad (\text{Glg. 1.4}),$$

wobei es sich bei \hat{R} um die räumliche Ausdehnung und bei \hat{T} um die zeitliche Ausdehnung des Wilson Loops handelt, gemessen in Einheiten des Gitterabstandes a . Die Integration erfolgt hier über alle Linkvariablen. Da Glg. 1.4 proportional zur Korrelationsfunktion ist (vgl. Glg. 1.1), entspricht sie im Limes großer euklidischer Zeiten Glg. 1.0. Man kann nun nach dem statischen Quark-Antiquark Potential $V_{Q\bar{Q}}(r)$ auflösen und es ergibt sich:

$$V_{Q\bar{Q}}(\hat{R}) = \lim_{\hat{T} \rightarrow \infty} \ln \left(\frac{\langle W(\hat{R}, \hat{T}) \rangle}{\langle W(\hat{R}, \hat{T}+1) \rangle} \right) \quad (\text{Glg. 1.5}),$$

wobei die Größe:

$$V_{eff}(\hat{R}, \hat{T}) = \ln \left(\frac{\langle W(\hat{R}, \hat{T}) \rangle}{\langle W(\hat{R}, \hat{T}+1) \rangle} \right) \quad (\text{Glg. 1.6})$$

als das effektive Potential bezeichnet wird.

Die numerische Bestimmung der Erwartungswerte von Wilson Loops erlaubt es somit, das statische Potential zu bestimmen.

1.1 Der on-axis Wilson Loop

Man betrachte noch einmal die Gitterversion des Wilson Loops:

$$W_C[U] = W(\hat{R}, \hat{T}) = \frac{1}{2} \text{Tr} \left(\prod_{l \in C} U_l \right) .$$

Im Falle eines on-axis Wilson Loops verläuft der räumliche Anteil des Wilson Loops, beschrieben durch \hat{R} , entlang bzw. auf einer der drei räumlichen Achsen der vierdimensionalen euklidischen Raumzeit:

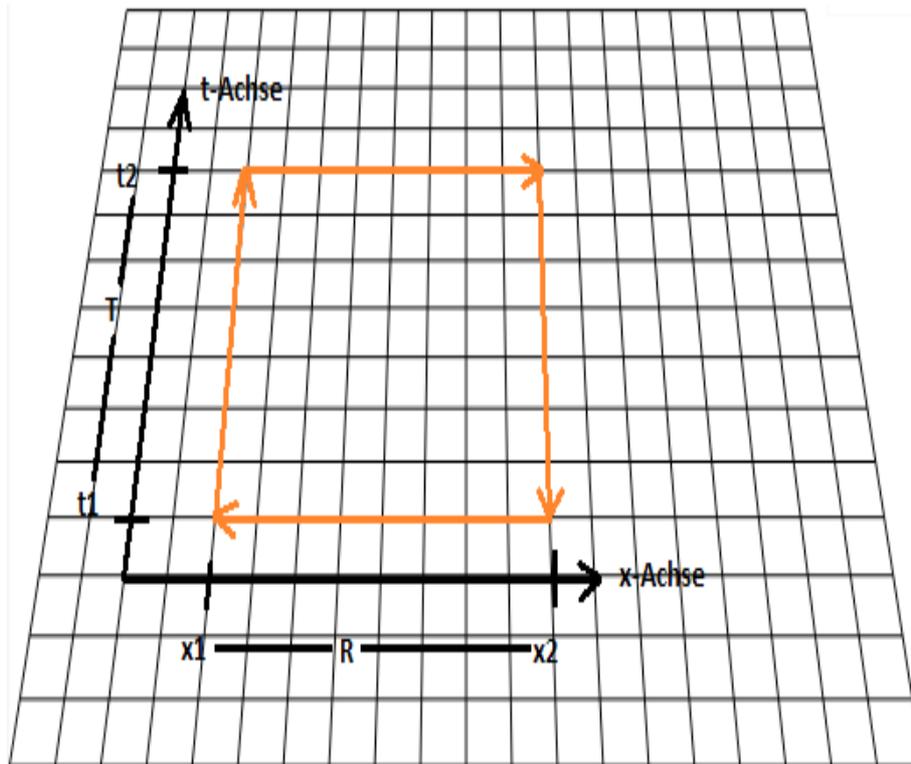


Abb. 1.0: Darstellung eines on-axis Wilson Loops

Jeder Gitterpunkt in Abb. 1.0 entspricht einem Raumzeitpunkt. Zwischen zwei direkt, axial benachbarten Gitterpunkten liegt ein Gitterlink, der einer Linkmatrix der SU(2) entspricht. Der Startpunkt des Wilson Loops ist der Raumzeitpunkt (t_1, x_1) . Zu Beginn werden die Zeitlinks miteinander multipliziert von (t_1, x_1) nach (t_2, x_1) , dann folgen die Raumlinks von (t_2, x_1) nach (t_2, x_2) , dann werden die adjungierten Zeitlinks von (t_2, x_2) nach (t_1, x_2) und zum Schluss die adjungierten Raumlinks von (t_1, x_2) nach (t_1, x_1) multipliziert. Der on-axis Wilson Loop sieht wie folgt aus:

$$\begin{aligned} W_C[U] &= W(\hat{R} = x_2 - x_1, \hat{T} = t_2 - t_1) \\ &= \frac{1}{2} \text{Tr} (U(x_1, t_1; x_1, t_2) U(x_1, t_2; x_2, t_2) U(x_2, t_2; x_2, t_1) U(x_2, t_1; x_1, t_1)) , \end{aligned}$$

wobei x_1, x_2, t_1, t_2 in Einheiten des Gitterabstandes a gegeben sind. Wie in der Abb. 1.0 zu sehen ist, wurde hier der on-axis Wilson Loop auf der x-Achse durchlaufen, wobei der Anfangspunkt und Endpunkt jeweils einem Gitterpunkt der x-Achse entsprechen müssen. Generell gilt (sowohl für on-axis als auch off-axis Wilson Loops), dass der Anfangspunkt der Position des Quarks und der Endpunkt (des räumlichen Anteils des Loops) der Position des Antiquarks entsprechen kann. Bei einem on-axis Wilson Loop werden somit räumliche Ausdehnungen realisiert bzw. gewählt die ganzzahlige Vielfache des Gitterabstandes sind, d.h. $\hat{R}=0,1,2,3,\dots$. Berücksichtigt man Glg. 1.5, so lassen sich daher mit on-axis Wilson Loops lediglich Potentialwerte für ganzzahlige Abstände zwischen Quark und Antiquark bestimmen. Die Zielsetzung dieser Arbeit liegt in der effizienten numerischen Bestimmung der Potentialwerte für nicht ganzzahlige räumliche Abstände ($\hat{R}=\sqrt{2},\sqrt{3},\sqrt{4},\sqrt{5},\dots$) bzw. für nicht-axiale Quark-Antiquark Separationen.

1.2 Der off-axis Wilson Loop

Für einen off-axis Wilson Loop kann die räumliche Ausdehnung beliebig gewählt werden. Die Raumzeitpunkte, die den Beginn und das Ende des räumlichen Anteils des Wilson Loops festlegen, müssen nicht beide auf eine räumliche Koordinatenachse gesetzt werden. Die folgende Abbildung Abb. 1.1 zeigt einen off-axis Wilson Loop für eine räumliche Trennung des Quark-Antiquark Paares entlang der Flächendiagonale der x-y Ebene:

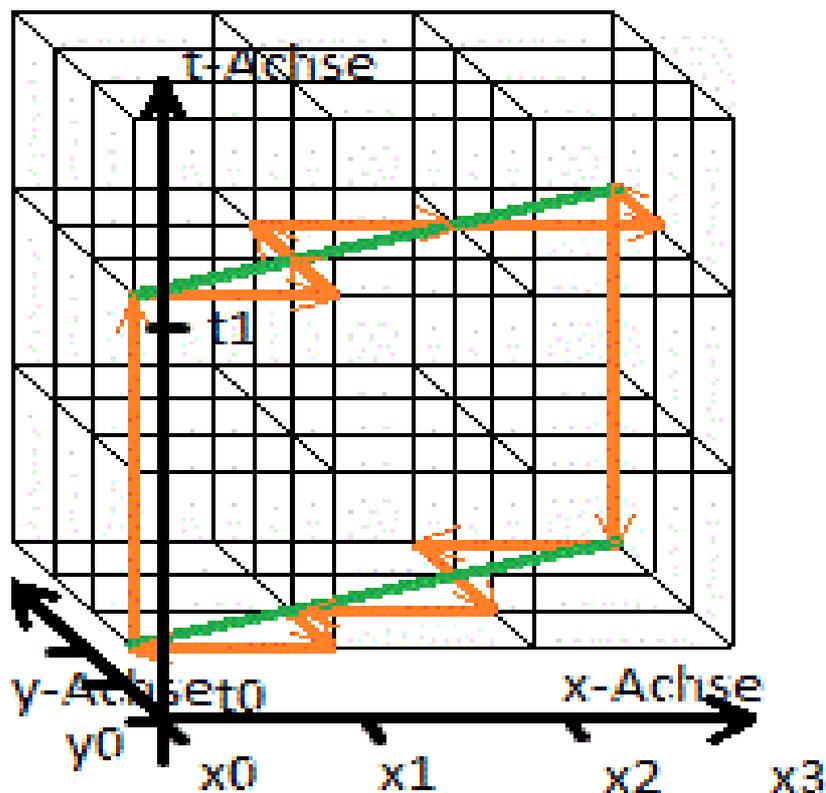


Abb. 1.1: Darstellung eines off-axis Wilson Loops

Am zeitlichen Anteil des off-axis Wilson Loops ändert sich nichts verglichen mit einem on-axis Wilson Loop. Wie bereits im Voraus erwähnt, werden die Zeitlinks entlang der

Zeitachse multipliziert. Für die Multiplikation der Raumlänge ergibt sich eine Änderung. Der räumliche Anteil des off-axis Wilson Loops muss möglichst dicht am grünen Pfad der Abb. 1.1 verlaufen. Im Allgemeinen muss auf dem Weg vom Beginn bis zum Ende des räumlichen Pfades über diejenigen Zwischenpunkte des Gitters gelaufen werden, die jeweils den geringsten Abstand zur direkten Verbindungslinie zwischen Quark und Antiquark aufweisen, durch diese Bedingung wird die nicht-axiale Trennung am Besten genähert. Der direkte Pfad (grüne Linie in Abb. 1.1) kann nicht durchlaufen werden, da dort keine Gitterlinks existieren. Gitterlinks gibt es nur zwischen den direkt, axial benachbarten Raumzeitpunkten. Der off-axis Wilson Loop der Abb. 1.1 zeigt, dass ein Wilson Loop Wert für einen nicht ganzzahligen Abstand berechnet wird:

$$W_c[U] = W(\hat{R} = \sqrt{(x_3 - x_0)^2 + (y_3 - y_0)^2}, \hat{T} = t_1 - t_0) \quad ,$$

wobei räumliche und zeitliche Koordinatenwerte in Einheiten des Gitterabstandes gemessen werden.

1.3 Die numerische Bestimmung des Erwartungswerts des Wilson Loops [1]

Man betrachte noch einmal Glg. 1.4, die Gitterversion des Erwartungswerts des Wilson Loops im Grundzustand:

$$\langle W_c[U] \rangle_{eukl} = \langle W(\hat{R}, \hat{T}) \rangle = \frac{\int DU W_c[U] \exp(-S_{YM}^{eukl}[U])}{\int DU \exp(-S_{YM}^{eukl}[U])}$$

In dieser Gleichung muss über alle möglichen Linkvariablen integriert werden. Da es sich hier um ein hochdimensionales Integral handelt, wird zur Auswertung in der Numerik ein Monte-Carlo-Algorithmus, hier speziell ein Heatbath-Algorithmus, verwendet. Dabei wird eine kleine repräsentative Submenge an Link(variablen) - bzw. Eichfeldkonfigurationen erstellt, auf der die Wahrscheinlichkeitsverteilung $\exp(-S_{YM}^{eukl}[U])$ realisiert ist. Auf Grundlage dieser Untermenge von N Eichfeldkonfiguration approximiert man Glg. 1.4 wie folgt:

$$\langle W(\hat{R}, \hat{T}) \rangle = \langle W_c[U] \rangle_{eukl} = \frac{\int DU W_c[U] \exp(-S_{YM}^{eukl}[U])}{\int DU \exp(-S_{YM}^{eukl}[U])} \approx \frac{1}{N} \sum_{i=1}^N W_c(\{U_i\}) \quad ,$$

wobei mit $\{U_i\}$ ($i = 1, \dots, N$) die i-te Link(variablen) - bzw. Eichfeldkonfiguration bezeichnet wird.

Ausführlich bedeutet dies, man legt zu Beginn einen Wilson Loop $W(\hat{R}, \hat{T})$ fest. Durch die Festlegung der räumlichen Ausdehnung des Wilson Loops ist auch bestimmt, ob es sich um einen on- oder off-axis Wilson Loop handelt. Ein on-axis Wilson Loop besitzt eine räumliche Ausdehnung entlang einer der Raumachsen und ein off-axis Wilson loop entlang einer beliebigen Geraden. Man betrachtet nun die erste Eichfeldkonfiguration $\{U_1\}$. Der Wilson Loop wird für jeden Startpunkt, d.h. Raumzeitpunkt der Eichfeldkonfiguration, ausgewertet. Zusätzlich wird an jedem Raumzeitpunkt für alle möglichen „Permutationen der räumlichen Ausrichtung des Wilson Loops“ (erläutert in Kapitel 2, Programm: Wilson_loop_averages_1_BA.cpp, Schritt 2) der Wilson Loop ausgerechnet. Alle Ergebnisse der Wilson Loop Auswertungen werden addiert. Zum Schluss wird durch die Anzahl der Raumzeitpunkte der Eichfeldkonfiguration und durch die

Zahl der möglichen Permutationen, die die räumliche Ausrichtung des Wilson loops zulässt, dividiert. Die Idee dieses Vorgehens besteht darin, den Wilson Loop über alle zeitlichen als auch räumlichen Linkvariablen der Eichfeldkonfiguration durchzuführen und zu mitteln. Gleiches Vorgehen gilt für alle weiteren Eichfeldkonfigurationen von $\{U_2\}$ bis $\{U_N\}$. Die gemittelten Wilson Loop Ergebnisse aus den jeweiligen Eichfeldkonfigurationen werden summiert und durch die Anzahl der Eichfeldkonfigurationen geteilt, somit handelt es sich um eine weitere Mittelung, in diesem Fall über alle Eichfeldkonfigurationen. Das hieraus folgende Ergebnis entspricht dem Erwartungswert des zu Beginn festgelegten Wilson Loops im Grundzustand.

2. Numerisches Konzept: Die Implementierung des off-axis Wilson Loops und die Effizienz

2.0 Erklärungen zur Implementierung des off-axis Wilson Loops

Der Funktionskopf zur Berechnung eines off-axis Wilson Loops (siehe `Wilson_loop_off_axis_1()` in `Wilson_loops.cc`) sieht wie folgt aus:

```
complex Wilson_loop_off_axis_1(double *gauge_field, int T, int L, int it, int ix, int iy, int iz,
int extt, int extx, int exty, int extz)
```

Dabei ist (it, ix, iy, iz) der Raumzeitpunkt in einer Eichfeldkonfiguration, an dem der off-axis Wilson Loop startet. Dabei sitzt das Quark am Ortspunkt (ix, iy, iz) und das Antiquark am Ortspunkt $(ix+extx, iy+exty, iz+extz)$. Der Vektor $(extx, exty, extz)$ startet am Ortspunkt des Quarks, endet am Ortspunkt des Antiquarks im räumlichen Anteil des Gitters und liegt somit parallel zur direkten Verbindungslinie zwischen Quark und Antiquark. Dieser Vektor legt die räumliche Ausdehnung des off-axis Wilson Loops fest. Setzt man diesen Vektor parallel zu einer der Raumachsen (, d.h.: $(1,0,0)$, $(0,1,0)$, $(0,0,1)$), so kann man mit der obigen Funktion einen on-axis Wilson Loop durchführen. Der zeitliche Anteil des off-axis Wilson Loops wird durch `extt` bestimmt. Die Parameter `gauge_field`, `T` und `L` beschreiben die Eichfeldkonfiguration, auf der der off-axis Wilson Loop durchgeführt wird. Der Ausgabewert wird hier als komplex beschrieben, aber im Rahmen der Yang-Mills-Theorie ergeben sich für den Wilson Loop reelle Ergebnisse zwischen -1 und 1. Auch nach der Mittelung eines „Wilson Loops in allen Permutationen der räumlichen Ausrichtung“ (erläutert im Nachfolgenden, Programm: `Wilson_loop_averages_1_BA.cpp`, Schritt 2) über alle Raumzeitpunkte ergibt sich ein reeller Wert zwischen -1 und 1. Erst nach der Mittelung des Wilson Loops über alle Eichfeldkonfigurationen erhält man einen rein positiven, reellen Wert.

Man betrachte nun den Algorithmus, der dieser off-axis Wilson Loop Funktion zugrunde liegt (vgl. `Wilson_loop_off_axis_1()` in `Wilson_loops.cc`):

Schritt 1: Der räumliche Anteil des eingegebenen Startpunkts (ix, iy, iz) wird in (x_i, y_i, z_i) gespeichert und der räumliche Anteil des Endpunktes $(ix+extx, iy+exty, iz+extz)$ wird in (x_f, y_f, z_f) gespeichert. Der Grund hierfür liegt darin, dass nach jeder Linkmultiplikation der Raumzeitpunkt (ix, iy, iz, it) sich verändert bzw. aktualisiert wird.

Schritt 2: Die Zeitlinks werden von (ix, iy, iz, it) nach $(ix, iy, iz, it+extt)$ miteinander multipliziert und das entstandene Linkmatrizenprodukt ist in der Matrix `double U[8]` gespeichert. In

jedem Schritt wurde der ursprüngliche Wert von it erhöht, sodass am Ende der Multiplikation die Variable it bzgl. ihres ursprünglichen Werts einen um $extt$ erhöhten Wert enthält. Man multipliziert als nächstes die Raumlinks.

Schritt 3: Es wird der erste temporäre Ortspunkt erstellt $(x_{tmp}, y_{tmp}, z_{tmp}) = (ix+1, iy, iz)$, man berechnet den Abstand d_1 zur direkten Verbindungslinie zwischen Quark und Antiquark. Die direkte Verbindungslinie kann im Rahmen der linearen Algebra mithilfe einer Geradengleichung beschrieben werden, bei der der Stützvektor (x_i, y_i, z_i) ist und der Richtungsvektor (ext_x, ext_y, ext_z) entspricht. Man betrachtet dann den zweiten, möglichen Ortspunkt $(ix, iy+1, iz)$ und berechnet auch hier d_2 . Für den dritten Ortspunkt $(ix, iy, iz+1)$ berechnet man d_3 . Man geht somit ausgehend vom aktuellen Ortspunkt (ix, iy, iz) jeweils nacheinander einen Schritt in die jeweiligen Raumachsenrichtungen und berechnet den Abstand zur Quark-Antiquark Geraden. Hier werden die Variablen ix , iy und iz noch nicht verändert.

Schritt 4: Man bestimmt den kleinsten Wert aus den drei Abstandswerten $\{d_1, d_2, d_3\}$. Ist beispielsweise d_1 (d_2 , d_3) der kleinste Abstandswert, so wird am derzeitigen Raumzeitpunkt ein Raumlink in x (y , z)-Achsenrichtung ausgewählt. Nach der Multiplikation mit dem jeweiligen Raumlink wird die x -, y - oder z -Koordinate um 1 erhöht bzw. der aktuelle Ortspunkt (ix, iy, iz) wird zu einem der temporären Ortspunkte. Die Schritte 3 und 4 werden solange wiederholt, bis der aktuelle Ortspunkt (ix, iy, iz) den Ortspunkt des Antiquarks (x_f, y_f, z_f) erreicht hat.

Schritt 5: Nun werden die adjungierten Zeitlinks miteinander multipliziert von $(ix, iy, iz, it+extt)$ bis (ix, iy, iz, it) . Die Variable it enthält nun wieder ihren ursprünglichen Wert.

Schritt 6: Das bisherige Matrizenprodukt ist in $double\ U[8]$ gespeichert. Das nun folgende Matrizenprodukt der Raumlinks wird in $double\ U1[8]$ gespeichert. Man startet wieder am Ortspunkt (x_i, y_i, z_i) , erstellt dann – so wie in Schritt 3 temporäre Ortspunkte mit den dazugehörigen Abständen d_1 , d_2 , d_3 . Danach folgt Schritt 4 mit der Bestimmung des kleinsten Abstandes, der Multiplikation des dazugehörigen Raumlinks und der Aktualisierung des Ortspunktes. Die Schritte 3 und 4 werden solange wiederholt bis der aktuelle Ortspunkt die Position des Antiquarks (x_f, y_f, z_f) erreicht hat.

Schritt 7: Das Matrizenprodukt in $U[8]$ wird mit dem adjungierten Matrizenprodukt in $U1[8]$ multipliziert, denn $U1[8]$ wurde von (x_i, y_i, z_i) nach (x_f, y_f, z_f) – somit entgegen der Pfadrichtung – erstellt. Die Raumlinkmultiplikation entgegen der Pfadrichtung mit dem anschließenden Adjungieren ist numerisch einfacher zu implementieren.

Schritt 8: Es erfolgt die Spurbildung und die Multiplikation mit dem Faktor $\frac{1}{2}$, sodass das Ergebnis des off-axis Wilson Loops vorliegt.

Man betrachte nochmals Schritt 3 und 4. Ausgehend von 3 temporären Ortspunkten sind 3 Abstandswerte entstanden. Es wird der kleinste Abstandswert aus $\{d_1, d_2, d_3\}$ bestimmt. Dann wird ein Raumlink in x , y oder z -Richtung für den aktuellen Raumzeitpunkt ausgewählt und der aktuelle Ortspunkt durch einen der temporären Ortspunkte ersetzt. Es kann dabei passieren, dass mehrere der temporären Ortspunkte den gleichen kleinsten Abstand haben. In derartigen Fällen wird einer der dazugehörigen Raumlinks bzw. einer der temporären Ortspunkte zufällig ausgewählt. Tatsächlich ist die Auswahl dieser temporären Ortspunkte nicht eindeutig, wie folgende Abbildung 2.0 veranschaulicht:

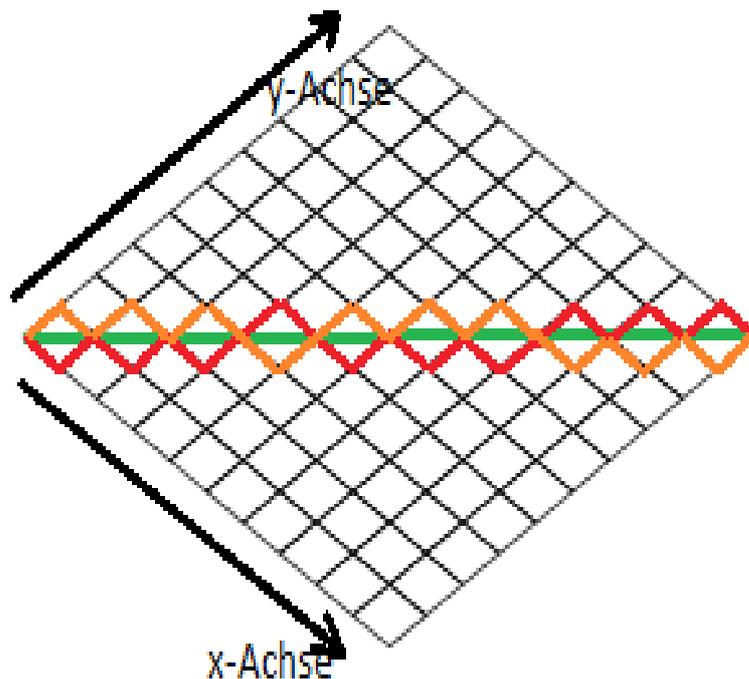


Abb. 2.0: Zwei mögliche räumliche Pfade eines off-axis Wilson Loops

Bei der Berechnung eines off-axis Wilson Loops kann für den räumlichen Anteil des Loops der gesamte orangene Pfad der Raumlinks durchlaufen werden. Jeder gewählte Ortspunkt im Raumgitter realisiert den kleinsten Abstand zur direkten Verbindungslinie (grüner Pfad). Dies gilt jedoch auch für den roten Pfad, der komplementär zum orangenen Pfad verläuft, somit könnte ein off-axis Wilson Loop im räumlichen Anteil auch den gesamten roten Pfad durchlaufen. Es gibt somit mehrere Möglichkeiten das Produkt der Raumlinks zu realisieren, trotz der Bedingung möglichst dicht entlang der Verbindungslinie zu bleiben. Dies kommt besonders auf dem Rückweg des räumlichen Anteils des Wilson Loops in Schritt 6 zur Geltung. Es wird nicht zwangsläufig der gleiche räumliche Pfad des Hinwegs aus den Schritten 3 und 4 einfach wieder zurückgelaufen. An Raumzeitpunkten, bei denen es mehrere, nächste, äquidistante Ortspunkte gibt, kann aufgrund der zufälligen Entscheidung ein anderer Ortspunkt ausgewählt werden, als dies auf dem Hinweg des Wilson Loops der Fall war. Dies garantiert, dass die kubische Symmetrie des Raumzeitgitters erhalten bleibt, da alle räumlichen Pfade gleichberechtigt auftreten können. Weiterhin wird dadurch garantiert, dass das statische Potential für ein Quark-Antiquark Paar im Grundzustand berechnet wird bzw. der Einfluss von höheren Energiezustände vermieden wird.

Die off-axis Wilson Loop Funktion wird auf die generierten Eichfeldkonfigurationen angewandt mit dem Ziel Erwartungswerte von Wilson Loops zu berechnen. Der Algorithmus dieses zweiten Programmcodes (siehe `Wilson_loop_averages_1_BA.cpp`) wird im Nachfolgenden erläutert:

Schritt 1: Man legt die möglichen Zeitausdehnungen der Wilson Loops mit der Variablen WL_T_max fest. Damit liegen alle möglichen Zeitausdehnungen $extt$ im Intervall: $0 \leq extt \leq WL_T_max$. Die möglichen, räumlichen Ausdehnungen von Wilson Loops legt die Variable WL_R_max fest. Es wird dabei ein Würfel der Kantenlänge WL_R_max im Ursprung im räumlichen Anteil des Raumzeitgitters definiert (vgl. Abb. 2.1: orangener Würfel). Die Vektoren, die im Ursprung beginnen und innerhalb oder auf der Oberfläche des Würfels enden, entsprechen möglichen räumlichen Ausdehnungen des Wilson Loops ($extx$, $exty$, $extz$) (vgl. Abb. 2.1: rote Vektoren).

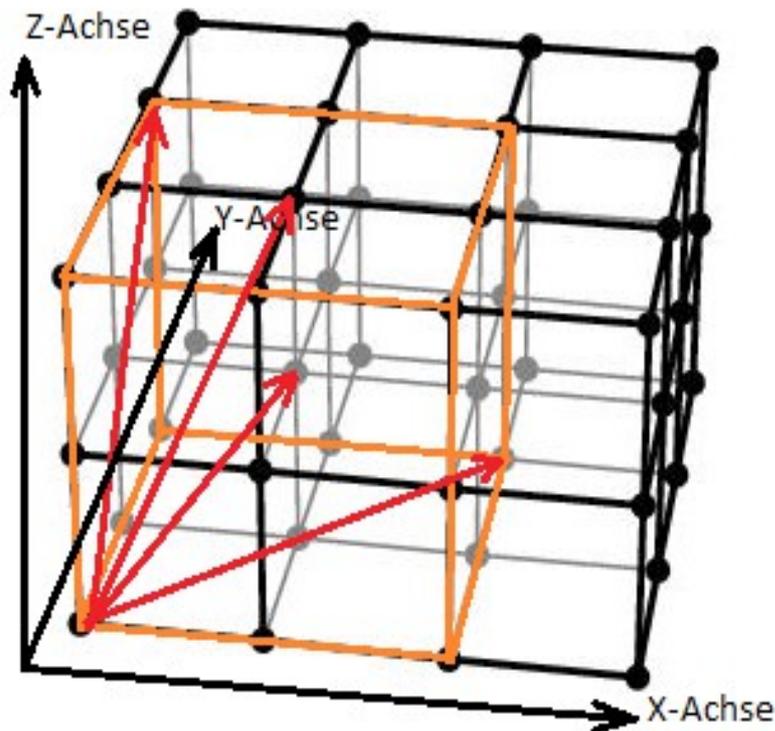


Abb. 2.1: Würfel mit verschiedenen räumlichen Ausdehnungen des Wilson Loops

Schritt 2: Die möglichen Vektoren ($extx$, $exty$, $extz$) im Würfel müssen vorher bestimmt und in einer Matrix M gespeichert werden. Es sind allerdings nicht alle möglichen Vektoren zu speichern, es genügt bereits einen kleinen Anteil auszuwählen. Die ausgelassenen Vektoren lassen sich nämlich durch Permutation der Koordinaten der gespeicherten Vektoren konstruieren. Beispielsweise lassen sich aus dem Vektor $(1,0,0)$ die Vektoren $(0,1,0)$ und $(0,0,1)$ durch Permutation der Koordinaten erzeugen. Aus dem Vektor $(1,2,3)$ lassen sich die Vektoren $(1,3,2)$, $(3,2,1)$, $(3,1,2)$, $(2,1,3)$ und $(2,3,1)$ erzeugen. Unter dem Ausdruck „... alle Permutationen der räumlichen Ausrichtung des Wilson Loops ...“, versteht man die Permutation der Koordinaten eines vorgegebenen Vektors ($extx$, $exty$, $extz$), sowie es in den obigen Beispielen gezeigt ist. Geometrisch bedeutet das Permutieren der Koordinaten, dass man an gewissen Ebenen spiegelt.

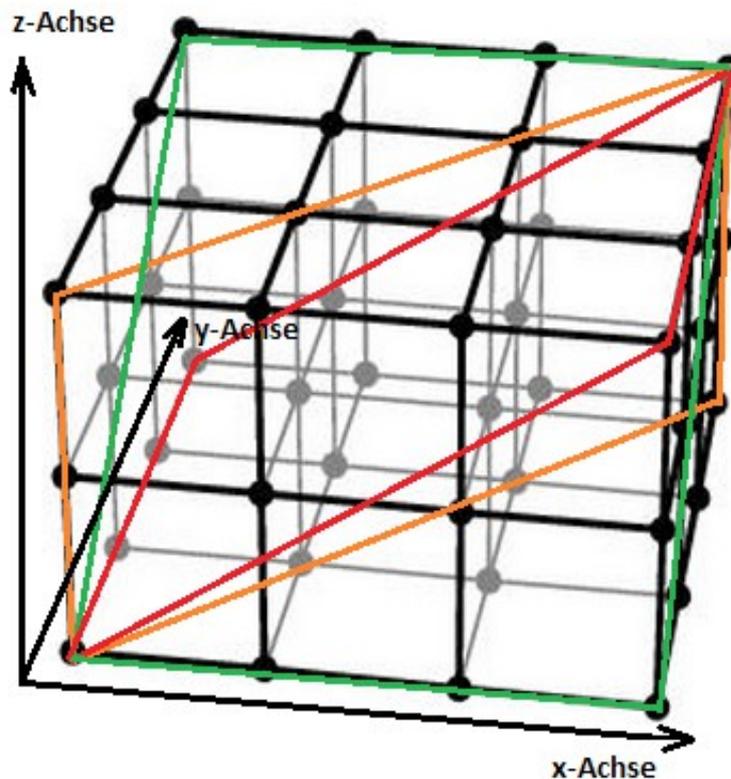


Abb. 2.2: Darstellung der Spiegelebenen im räumlichen Anteil des Gitters

Diese Ebenen sind in der obigen Abbildung gezeigt. Die orange, umrandete Ebene wird aufgespannt durch die Winkelhalbierende (zwischen der x- und y-Achse) und durch die z-Achse. Eine Spiegelung an dieser Ebene entspricht einer Permutation der x und y Koordinaten. Mit anderen Worten es genügt also die Punkte, die auf einer Seite dieser orangenen Ebene liegen zu speichern, da die restlichen durch eine Permutation erreichbar sind. Damit reduziert sich der Umfang der zu speichernden Vektoren um die Hälfte. Die weiteren Ebenen reduzieren die Anzahl der zu speichernden Vektoren zusätzlich. Es müssen nur noch die Ortspunkte im räumlichen Anteil des Gitters gespeichert werden, die in der folgenden Abb. 2.3 im rot eingerahmten Bereich liegen.

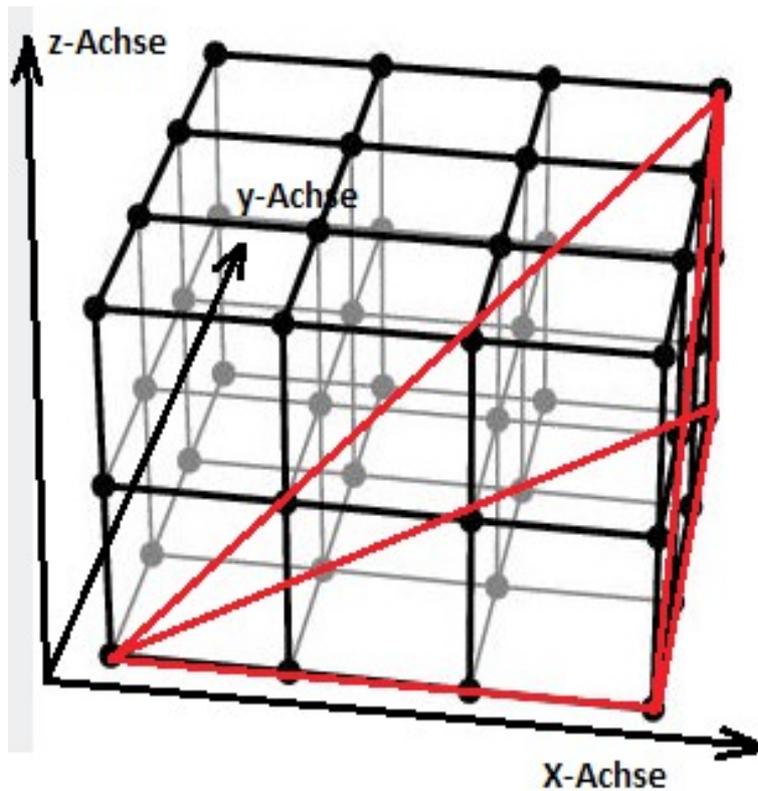


Abb. 2.3: Darstellung der zu speichernden Ortspunktemenge

Schritt 3: Man wählt einen Vektor (ext_x, ext_y, ext_z) aus der Matrix M aus und legt eine zeitliche Ausdehnung ext_t des Wilson Loops fest. An jedem Raumzeitpunkt wird dieser Wilson Loop in allen Permutationen von (ext_x, ext_y, ext_z) ausgeführt. Alle Wilson Loop Werte werden summiert und zum Schluss durch die Zahl der Gitterpunkte und die Zahl der möglichen Permutationen von (ext_x, ext_y, ext_z) geteilt. Dieser gemittelte Wilson Loop Wert wird in einer Matrix WL für die zeitliche Ausdehnung ext_t und die räumliche Ausdehnung (ext_x, ext_y, ext_z) gespeichert. Diese Prozedur wird für alle Vektoren der Matrix M und für alle möglichen Zeitausdehnungen ext_t durchgeführt. Wichtig anzumerken ist, dass eine Permutation der Koordinaten eines Vektors (ext_x, ext_y, ext_z) keine neue räumliche Ausdehnung beschreibt. Es handelt sich dabei stets um denselben Wilson Loop. So entsprechen beispielsweise die Vektoren $\{(4,0,1);(4,1,0);(1,4,0);(1,0,4);(0,4,1);(0,1,4)\}$ derselben räumlichen Ausdehnung eines off-axis Wilson Loops. Man kann nämlich alle sechs Vektoren mithilfe von Spiegelungen an den Ebenen der Abb. 2.2 ineinander überführen. Bei den Ebenenspiegelungen handelt es sich um Symmetrietransformationen des kubischen Raumgitters. Der Vektor $(3,2,2)$ hingegen und seine dazugehörigen Permutationen stellen eine andere räumliche Ausdehnung dar als der Vektor $(4,0,1)$, obwohl in beiden Fällen $R = \sqrt{3^2 + 2^2 + 2^2} = \sqrt{4^2 + 0^2 + 1^2} = \sqrt{17}$. Entscheidend ist jedoch, dass es keine Möglichkeit gibt, über Spiegelungen an den Ebenen der Abb. 2.2 diese Vektoren zur Deckung zu bringen. Daher werden Vektoren wie $(3,2,2)$ und $(4,0,1)$ in Schritt 3 in getrennten Prozeduren behandelt und getrennt gespeichert.

2.1 Betrachtung von effizienzsteigernden Methoden

Die grundlegende Idee der Effizienzsteigerungen liegt darin, wiederholt berechnete Zeitlink – bzw. Raumlinkprodukte zwischenspeichern oder in einer anderen Reihenfolge auszurechnen, um Rechenzeit einzusparen.

Man betrachte zuerst die Methode zur Einsparung von Raumlinks (siehe `Wilson_loop_averages_2_BA.cpp`). Betrachtet man nochmal die Vektoren (ext_x, ext_y, ext_z) in der Matrix M und nimmt an, dass diese Vektoren im Ursprung starten, so gibt es Teilmengen von Vektoren, deren Endpunkte auf gemeinsamen Geraden liegen. So liegen beispielsweise die Vektoren $(5,4,0)$ und $(10,8,0)$ auf einer Geraden, wie die Abb. 2.4 unten gezeigt. Unter Ausnutzung der Eigenschaft, dass beide Vektoren auf einer gemeinsamen Geraden liegen, lassen sich zwei Wilson Loop Werte zügig ermitteln. Bei der Multiplikation der Raumlinks (grüner Pfad) bis $(5,4,0)$ kann das Matrizenprodukt zwischengespeichert werden und man setzt mit der Raumlinkmultiplikation fort (orangener Pfad) bis $(10,8,0)$. Würde man nicht zwischenspeichern, so müsste vom Ursprung bis $(10,8,0)$ das Raumlinkprodukt völlig neu gebildet werden. Man spart somit die Rechenzeit für die Raumlinkmultiplikation bis $(5,4,0)$ ein.

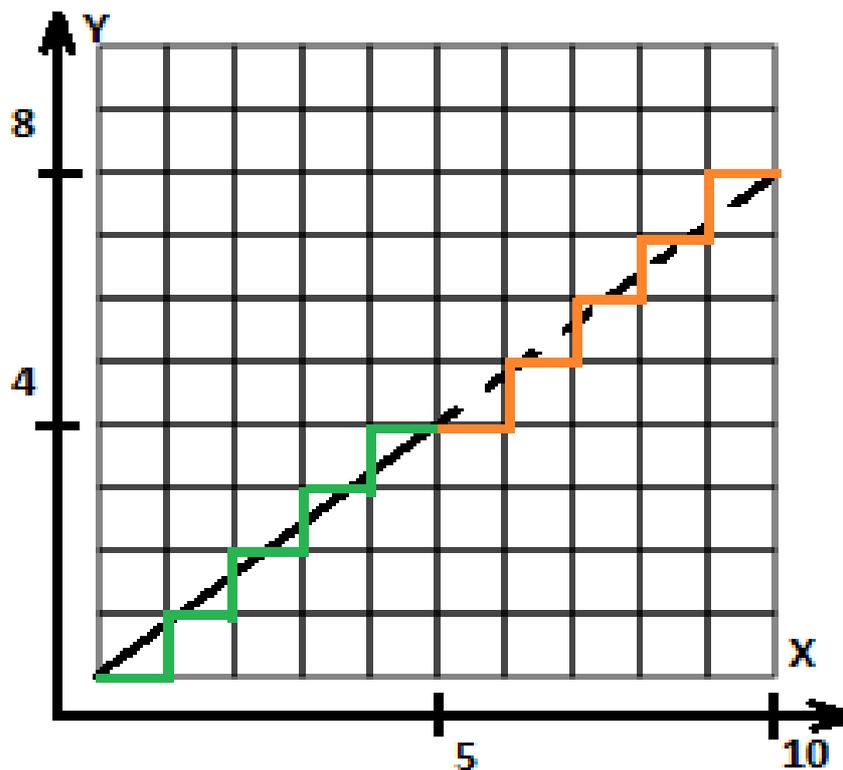


Abb. 2.4: Fortsetzen eines Raumlinkproduktes entlang einer Geraden

Als nächstes betrachte man die Methode zur Einsparung von Zeitlinks (siehe `Wilson_loop_averages_3_BA.cpp`). Bei der Berechnung eines Wilson Loops über alle Raumzeitpunkte wird an jedem Raumzeitpunkt das entsprechende Zeitlinkprodukt gebildet. Jedoch wird für alle Wilson Loops mit einer anderen räumlichen Ausdehnung und

mit der gleichen zeitlichen Ausdehnung, dasselbe Zeitlinkprodukt wiederholt an allen Raumzeitpunkten erzeugt. Legt man zu Beginn eine zeitliche Ausdehnung für alle Wilson Loops fest, berechnet an allen Raumzeitpunkten das Zeitlinkprodukt für diese zeitliche Ausdehnung und speichert dies. So kann man die Rechenzeit für eine wiederholte Berechnung der Zeitlinks einsparen, indem das benötigte Zeitlinkprodukt für den jeweiligen Raumzeitpunkt einfach aus einem Speicher abgerufen wird.

		Wilson_loop_averages_1_BA.cpp	Wilson_loop_averages_2_BA.cpp	Wilson_loop_averages_3_BA.cpp	Wilson_loop_averages_4_BA.cpp
		Ohne Effizienzsteigerungen	Mit Raumlink-Effizienz	Mit Zeitlink-Effizienz	Mit Raum- und Zeitlink-Effizienz
Laufzeit	real	31m 10.117s	27m 22.743s	26m 29.963s	22m 48.385s
	user	16m 24.344s	14m 43.184s	12m 40.500s	11m 37.012s
	sys	0m 9.748s	0m 7.400s	0m 8.912s	0m 6.960s

Implementiert man die Methoden der Raumlink- bzw. Zeitlinkeinsparung, so ergibt sich eine grundlegende Verkürzung der Programmlaufzeiten (vgl. obige Tabelle). In jeder der vier genannten Programme wurde eine einzige Eichfeldkonfiguration ausgewertet. Diese Eichfeldkonfiguration entspricht dabei einem Raumzeitgitter mit der räumlichen Ausdehnung $L = 16$, der zeitlichen Ausdehnung $T = 16$ und einem Gitterabstand $a = 0,102$ fm bzw. $\beta = 2,40$ [4]. Bei der Auswertung der Eichfeldkonfiguration wurden $WL_T_max = 10$ und $WL_R_max = 6$ gesetzt. Jeder der damit möglichen Wilson Loops wurde an jedem Raumzeitpunkt der Eichfeldkonfiguration und in allen Permutationen ausgeführt, summiert und durch die Zahl der Raumzeitpunkte und Permutationen geteilt. Durch Hinzunahme der Raum- und Zeitlinkeinsparung konnte die reale Laufzeit des Programms (Wilson_loop_averages_1_BA.cpp) um ca. 27% gesenkt werden. Wichtig anzumerken ist, dass das Programm in einer virtuellen Maschine mit einem Hauptspeicher von 2048 MB und mit einem CPU ausgeführt wurde.

Das effizienteste Programm (Wilson_loop_averages_4_BA.cpp) könnte erweitert werden, sodass eine Anwendung sogar in der QCD denkbar wäre. Es handelt sich dabei um eine nicht-abelsche Feldtheorie, die invariant ist unter lokalen Transformationen der Gruppe $SU(3)$ [2]. Man betrachte dazu das Programm Wilson_loop_averages_4_BA.cpp unter Berücksichtigung des Speicheraufwandes. Eine Eichfeldkonfiguration der QCD kann eine zeitliche Ausdehnung von $T = 64$ und eine räumliche Ausdehnung von $L = 32$ haben. Besonderen Speicheraufwand verlangt das time_field, was zum Speicher aller Zeitlinks dient ($32 \times 32 \times 32 \times 64 \times 1 \times 18 \times 8$ Bytes = 0,3 GB), das gauge_field, worin alle Links der Eichfeldkonfiguration gespeichert sind ($32 \times 32 \times 32 \times 64 \times 4 \times 18 \times 8$ Bytes = 1,2 GB) und das smeared_gauge_field_1, was alle Links der Eichfeldkonfiguration nach einem APE- und HYP-Smearing enthält ($32 \times 32 \times 32 \times 64 \times 4 \times 18 \times 8$ Bytes = 1,2 GB). Insgesamt hat das Programm Wilson_loop_averages_4_BA.cpp einen Speicherbedarf von 2,7 GB, dies lässt sich mit heutigen Arbeitsspeichergrößen von 8 – 16 GB bewältigen.

2.3 Weitere Methoden zur Effizienzsteigerung

Eine weitere Methode zur Effizienzsteigerung besteht darin, dass man einmalig eine Eichtransformation auf die Eichfeldkonfigurationen anwendet. Diese führt dazu, dass alle Zeitlinks zu Identitäten werden. Die Information, die ursprünglich in den Zeitlinks enthalten war, wird in die Raumlinks verschoben. Die Berechnung der Zeitlinks fällt damit vollständig weg und würde eine enorme Beschleunigung des Programms herbeiführen. Mit solchen Eichtransformationen kann man natürlich nicht alle Raum- und Zeitlinks auf Identitäten setzen, es ist nur möglich die Information eines Wilson Loops innerhalb des Loops zu verschieben.

In allen off-axis Wilson Loop Funktionen wird in den Schritten 3 und 4 zuerst eine Abstandsbestimmung für 3 temporäre Ortspunkte mit den Methoden der linearen Algebra durchgeführt, dann wird der kleinste Abstand daraus ermittelt. Es wird dabei mit double Zahlen gearbeitet. Eine Möglichkeit die Rechenzeit zu reduzieren besteht darin, den Bresenham-Algorithmus zu verwenden. Dieser Algorithmus arbeitet mit Integer Zahlen und braucht weniger Rechenoperationen, um den räumlichen Pfad zu bestimmen.

3. Ergebnisse und Diskussion: Das statische Quark-Antiquark Potential für nicht-axiale Separationen

3.0 Beschreibung der Graphen zum statischen Potential

Die nachfolgenden Abbildungen sind durch eine Auswertung von 15 Eichfeldkonfigurationen entstanden. Alle Eichfeldkonfigurationen hatten eine zeitliche Ausdehnung von $T = 16$, eine räumliche Ausdehnung von $L = 16$ und einen Gitterabstand von $a = 0,102$ fm bzw. $\beta = 2,40$ [4].

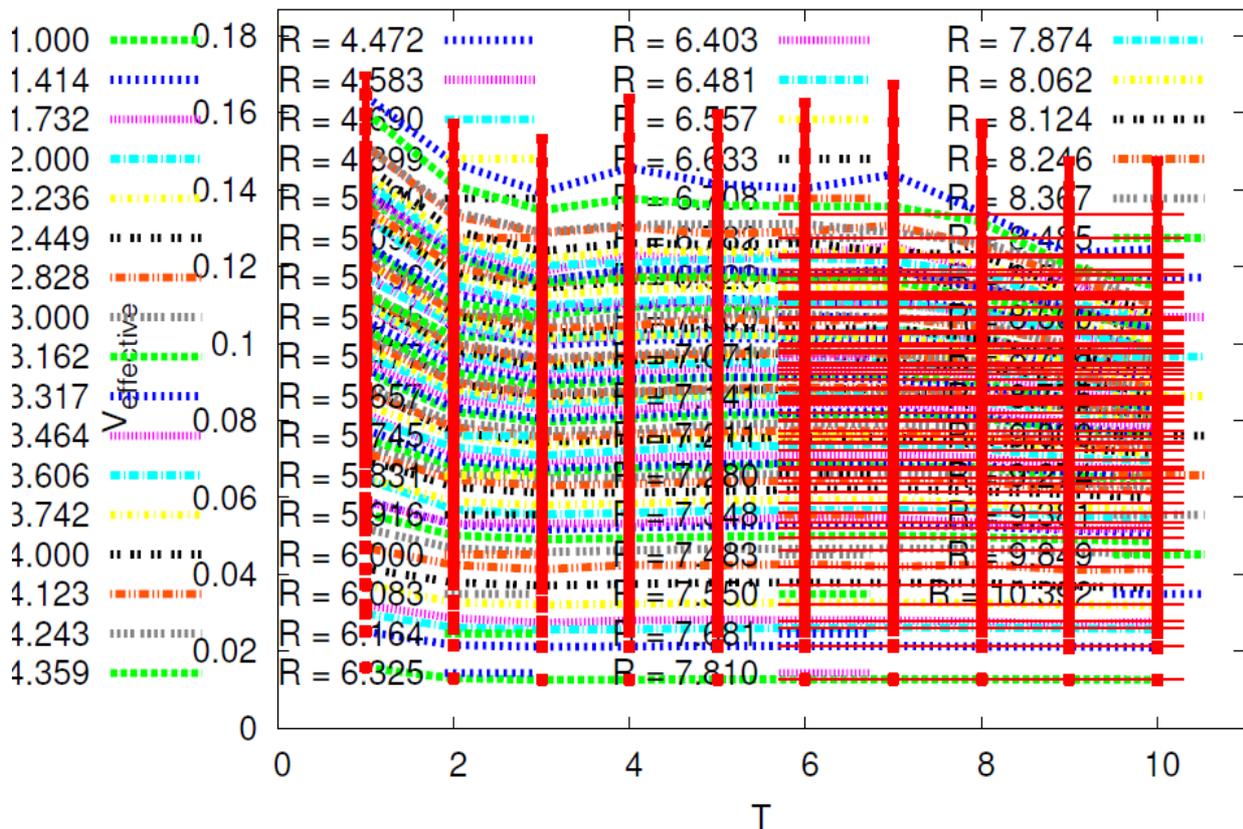


Abb. 3.0: Darstellung des effektiven Potentials gegen die Zeit für unterschiedliche räumliche Ausdehnungen R im off-axis Fall

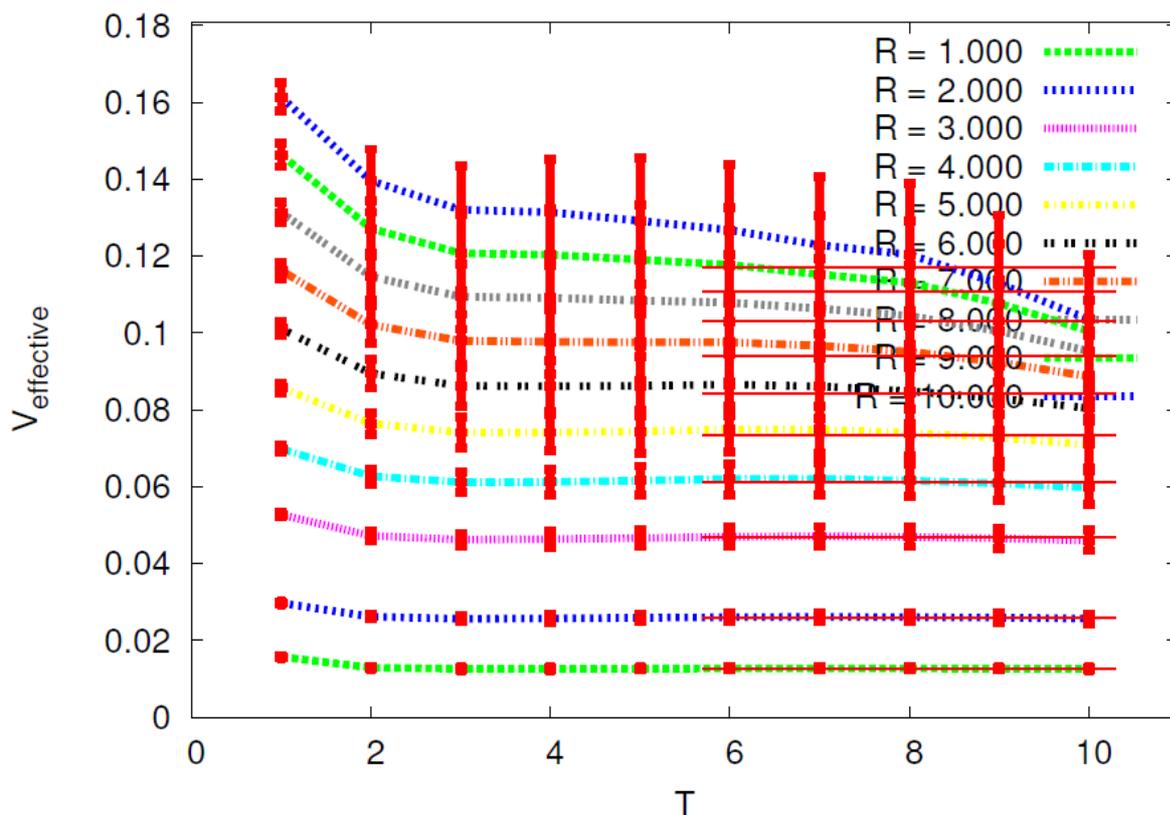


Abb. 3.1: Darstellung des effektiven Potentials gegen die Zeit für unterschiedliche räumliche Ausdehnungen R im on-axis Fall

Die obige Abbildung zeigt das effektive Potential (vgl. Glg. 1.6) für das Quark-Antiquark Paar. Es wurde V_{eff} gegen die zeitliche Ausdehnung \hat{T} aufgetragen für unterschiedliche räumliche Ausdehnungen \hat{R} . Man erkennt, dass ab einem gewissen Zeitpunkt das effektive Potential sich auf einem Plateau mit einem konstanten Potentialwert stabilisiert hat. Dieses Verhalten wird durch Glg. 1.0 beschrieben. Denn mit größer werdender Zeit werden Beiträge von höheren Energiezuständen unterdrückt und es dominiert der Beitrag des Grundzustandes. Zum Vergleich wurde das on-axis und off-axis Ergebnis untereinander aufgeführt.

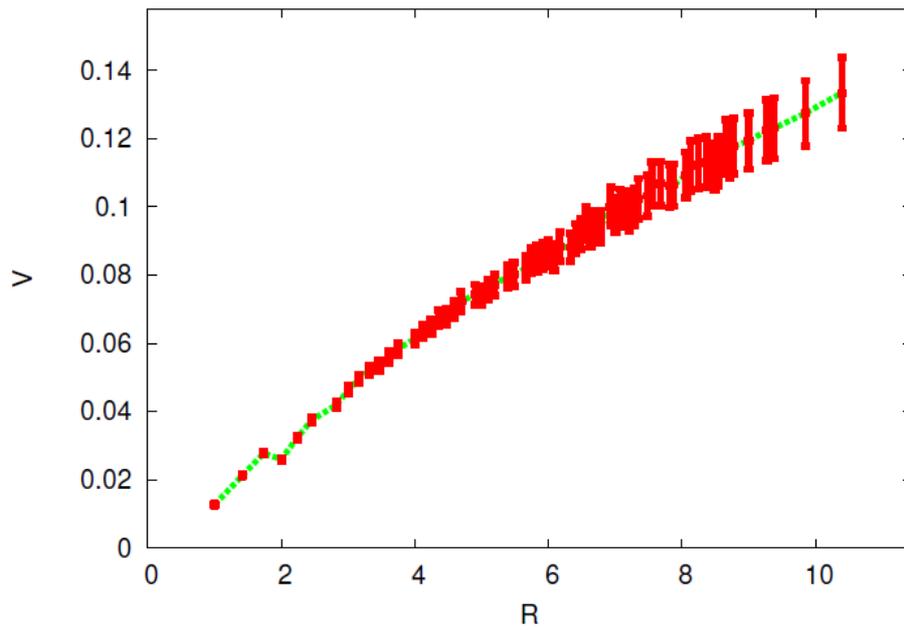


Abb. 3.2: Darstellung des statischen Potentials $V(R)$ für nicht-axiale Quark-Antiquark Separationen

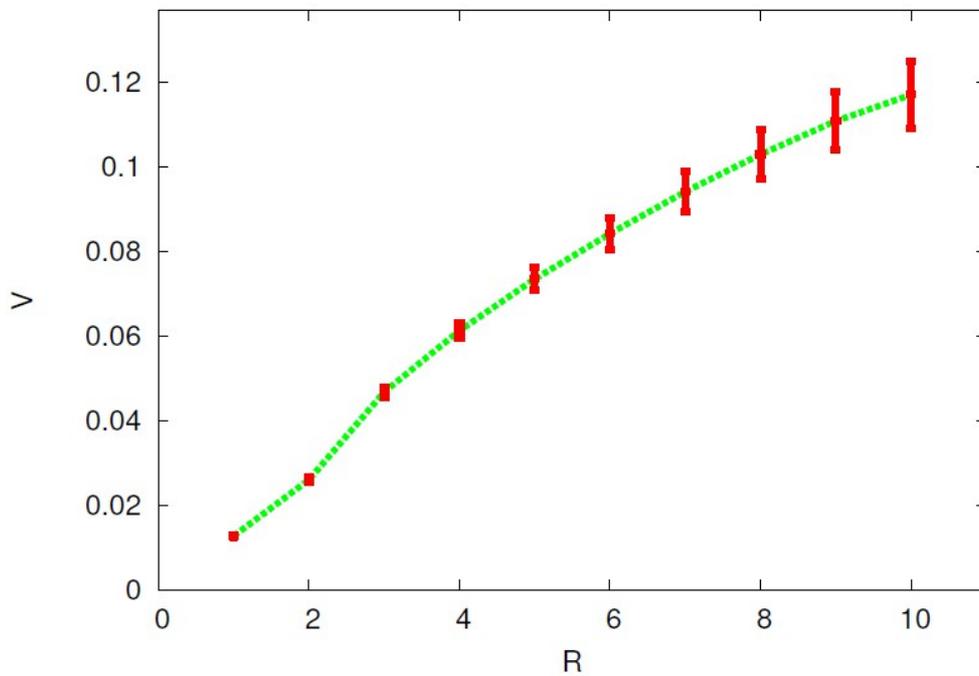


Abb. 3.3: Darstellung des statischen Potentials $V(R)$ für axiale Quark-Antiquark Separationen

Die obigen Abbildungen zeigen das statische Potential V für ganzzahlige und nicht-ganzzahlige Quark-Antiquark Abstände. Man erkennt in beiden Fällen den linearen Anstieg des Potentials für großer werdende Abstände zwischen Quark und Antiquark. Die Abbildung 3.3 verfügt lediglich die Auskunft über Potentialwerte für Abstände des Quark-Antiquark Paares, die ein ganzzahliges Vielfaches des Gitterabstandes sind. Diese \hat{R} - Werte entsprechen axialen Separationen. Die Abbildung 3.2 hingegen stellt Potentialwerte zusätzlich für nicht-ganzzahlige Vielfache des Gitterabstandes zur Verfügung. Diese entsprechen nicht-axialen Separationen zwischen Quark und Antiquark. Auffällig in der Abbildung Abb. 3.2 ist eine Knickstelle der Potentialkurve bei $\hat{R} = 2$. Der Grund hierfür liegt darin, dass es sich um einen Diskretisierungsfehler im Gitter handelt [3]. Besonders bei kurzen Abständen unterscheidet sich die Approximation der Quark-Antiquark Separation im Gitter am stärksten vom Kontinuum. Bei kurzen Abstände, wie bei $\hat{R} = 2$, wird die Rotationsinvarianz am stärksten verletzt. Natürlich gibt es auch Abweichungen bzw. Knickstellen bei größeren Separationen in der Potentialkurve der Abb 3.2, diese sind jedoch vergleichsweise klein und somit am Graphen kaum zu erkennen. Betrachtet man jedoch Potentialwerte für das Quark-Antiquark Paar für räumliche Abstände, die lediglich das (z.B.) $\sqrt{2}$ -fache oder lediglich das (z.B.) $\sqrt{3}$ -fache des Gitterabstandes sind, so ist die Knickstelle nicht mehr erkennbar, vielmehr liegt ein glatter Kurvenverlauf vor. Diesen glatten Kurvenverlauf weist auch Abbildung 3.3 auf, wo nur Potentialwerte für ganzzahlige Vielfache des einfachen Gitterabstandes gezeigt werden.

3.1 Qualitative Betrachtung der Ergebnisse

Das Ziel ist es zu überprüfen, ob der Gitterabstand - errechnet aus den Fitparametern des statischen Potentials - übereinstimmt mit dem vorgegebenen Gitterabstand bei der Erstellung der Eichfeldkonfigurationen mit $a = 0,102 \text{ fm}$. Dies gibt Auskunft über die Qualität der erstellten Potentialkurve für nicht-axiale Quark-Antiquark Separationen.

Ausgehend von den Fitparametern der Potentialkurven lässt sich zuerst eine Gitternäherung für den sog. Sommer – Parameter [5] bestimmen r_0^{Gitter} . Aus experimentellen Messungen ist bekannt, dass folgendes gilt:

$$F(r_0)r_0^2 = \left. \frac{dV}{dr} \right|_{r_0} r_0^2 = 1,65 \quad , \quad (\text{Glg. 3.1})$$

wobei $r_0 = 0,46 \text{ fm}$ der experimentell bestimmte Sommer-Parameter ist. Das Potential lässt sich für sehr kleine Entfernungen mit einem Coulomb-Term beschreiben, für größere Entfernungen hingegen dominiert der lineare Term

$$V = V_0 + \frac{\alpha}{r} + \sigma r \quad (\text{Glg. 3.2}).$$

Man betrachte die Ableitung des Potentials in Glg. 3.2 nach dem Ort r :

$$\frac{dV}{dr} = -\frac{\alpha}{r^2} + \sigma \quad (\text{Glg 3.3}).$$

Setzt man Glg. 3.3 in Glg. 3.1 ein, so lässt sich die Gitternäherung des Sommer-

Parameters wie folgt bestimmen:

$$\sigma - \frac{\alpha}{r_0^2} \cdot (r_0^{\text{Gitter}})^2 = 1,65$$
$$\Leftrightarrow r_0^{\text{Gitter}} = \sqrt{\frac{1,65 + \alpha}{\sigma}}$$

Ausgehend von den Fitparametern für σ und α erhält man als Gitternäherung des Sommer-Parameters:

$$r_0^{\text{Gitter}} = 4,451676.$$

Zur Bestimmung des Gitterabstandes rechnet man nun wie folgt:

$$a = \frac{r_0}{r_0^{\text{Gitter}}} = \frac{0,46 \text{ fm}}{r_0^{\text{Gitter}}} \quad (\text{Glg. 3.2})$$

Glg. 3.2 führt auf einen Gitterabstand von

$$a = 0,104 \text{ fm}$$

Der aus den Fitparametern bestimmte Gitterabstand weicht um ca. 2 % vom tatsächlich gewählten Gitterabstand bei Erstellung der Eichfeldkonfigurationen ab. Dies bestätigt eine gute Qualität der erstellten Potentialkurven für nicht-axiale Quark-Antiquark Separationen.

3.2 Abschließende Gesamtaussage

Das statische Quark-Antiquark Potential für nicht-axiale Separationen kann mithilfe des erstellten Programmcodes mit einer guten Qualität erfasst werden, was durch die Überprüfung des Gitterabstandes aus den Fitparametern bestätigt wurde. Bei der Effizienzsteigerung des Programmcodes konnte eine Reduktion der Rechenzeit von ca. 27 % erzielt werden. Weitere Maßnahmen, wie die Eichtransformation zur Identitätssetzung aller Zeitlinks oder aber der Bresenham-Algorithmus zur zügigen Bestimmung des räumlichen Pfades, ermöglichen weitere Einsparungen in der Rechenzeit. Eine Ausdehnung des Programmcodes auf die QCD, einer lokal SU(3) – invarianten Feldtheorie, ist ohne Schwierigkeiten erreichbar.

Literaturverzeichnis

- [1] : H. J. Rothe. *Lattice Gauge Theories – An Introduction*. World Scientific Lecture Notes in Physics - Vol. 59, 1997.
- [2] : D. H. Rischke: *Quantenfeldtheorie*, Vorlesungsskript, http://th.physik.uni-frankfurt.de/~drischke/Skript_QFT_I.pdf (2014).
- [3] : K. Jansen et al. $\Lambda_{\overline{MS}}$ - *from the static potential for QCD with $n_f = 2$ dynamical quark flavors*. arXiv:1110.6859v3 [hep-ph] (2011).
- [4] : O. Philipsen and M. Wagner. *On the definition and interpretation of a static quark anti-quark potential in the colour-adjoint channel*. arXiv:1305.5957v1 [hep-lat] (2013).
- [5] : A. Peters. *Determination of $\Lambda_{\overline{MS}}$ from the static quark-antiquark potential in momentum space*. Master Thesis, Goethe-Universität Frankfurt am Main (2014).

Hinweise zum Umgang mit dem Programmpaket

Ordner „lattice_SU2“: Dieser Ordner beinhaltet alle zur Auswertung einer Eichfeldkonfiguration nötigen Programme. Im Unterordner „Wilson_loop_averages“ befinden sich 4 Programme:

- 1.)Wilson_loop_averages_1_BA.cpp: Auswertung einer Eichfeldkonfiguration ohne Effizienzsteigerungen
- 2.)Wilson_loop_averages_2_BA.cpp: Auswertung einer Eichfeldkonfiguration mit Raumlinkeinsparung
- 3.)Wilson_loop_averages_3_BA.cpp: Auswertung einer Eichfeldkonfiguration mit Zeitlinkeinsparung
- 4.)Wilson_loop_averages_4_BA.cpp: Auswertung einer Eichfeldkonfiguration mit Zeit - und Raumlinkeinsparung

Jede Wilson_loop_averages Datei hat ihre eigene off-axis Wilson Loop Funktion:

Wilson_loop_off_axis_1,2,3,4()

Den Funktionskopf jeder off-axis Wilson Loop Funktion findet man in Wilson_loops.hh. Die vollständige Programmierung jeder off-axis Wilson Loop Funktion ist in Wilson_loops.cc vorzufinden.

Es existieren einige, kleinere Hilfsfunktionen, deren gesamter Programmcode in Wilson_loops.cc, der Funktionskopf aber in Wilson_loops.hh gespeichert worden ist:

- | | | | |
|--------------------|--------------|--------------------|------------------|
| 1.)mergesort_1() | 2.)zufall2() | 3.)permutation_2() | 4.)mergesort_2() |
| 5.)permutation_1() | 6.)zufall3() | 7.)permutator() | |

Im Unterordner „MC_heatbath“ befinden sich 15 Konfigurationsdateien, ein Ordner „WL_N_on_axis“ (enthält 15 WL_N-Dateien nach einer on-axis Auswertung der Eichfeldkonfigurationen) und ein Ordner „WL_N_off_axis“ (enthält 15 WL_N-Dateien nach einer off-axis Auswertung der Eichfeldkonfigurationen).

Ordner „PotentialQQ“: Im Unterordner „convert“ befindet sich ein Programm convert_BA.cpp. Dieses Programm dient speziell dazu, WL_N-Dateien, die aus off-axis Auswertungen entstanden sind, zu einer einzigen Datei correlations_matrices.dat umzuwandeln.

Selbstständigkeitserklärung

Hiermit versichere 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.

Frankfurt am Main, den 05.12.14

Sumir Motreedja