

# 3. Realisierung der Simulationen

Im Rahmen der Valenzquark-Approximation sind die fermionischen Felder keine dynamische Größen. Fermionische Propagatoren werden mit Hilfe geeigneter Quellen auf einer Eichfeldkonfiguration berechnet. Die Methoden zur Generierung von SU(3) Eichfeldkonfigurationen mit Hilfe von Monte-Carlo-Simulationen werden im folgenden Abschnitt beschrieben. Die verbleibenden Abschnitte dieses Kapitels befassen sich mit der Erzeugung von Quarkpropagatoren.

## 3.1. Generierung von Eichfeldern

In der Valenzquark-Approximation besteht das Problem der Generierung von SU(3) Eichfeldkonfigurationen  $\{U\}^{(i)}$  darin, dass diese nach Erreichen des Gleichgewichts mit einer Wahrscheinlichkeit

$$p_i \propto e^{-S_g(\{U\}^{(i)})} \quad (3.1)$$

zu erzeugen sind. Die Eichfeldwirkung  $S_g$  ist in Gleichung (2.8) definiert. Eine derartige Wahrscheinlichkeitsverteilung läßt sich mit Hilfe von Markov-Ketten im Rahmen einer Monte-Carlo-Simulation realisieren.

Zur Erzeugung von Eichfeldkonfigurationen mit Hilfe von Markov-Ketten müssen wir sicherstellen, dass die Wahrscheinlichkeitsverteilung  $p_i$  und die Übergangswahrscheinlichkeiten  $w_{ij}$  zwischen zwei Konfigurationen  $\{U\}^{(i)}$  und  $\{U\}^{(j)}$  folgende Bedingungen erfüllen:

1. Die Übergangswahrscheinlichkeiten müssen normiert sein, so dass für alle  $i$  gilt  $\sum_j w_{ij} = 1$ .
2. Falls  $p_i > 0$  und  $p_j > 0$  ist, so muss für ein endliches  $N > 0$  auch  $\sum_{\{i_k\}} w_{i_1 i_2} \dots w_{i_{N-1} i_N} > 0$  sein.
3. Die Übergangswahrscheinlichkeiten müssen die folgende Fixpunkteigenschaft erfüllen:  $\sum_i p_i w_{ij} = p_j$ .

Für die Erfüllung der dritten Bedingung ist es hinreichend, wenn folgende Forderung erfüllt ist, die als „detailed balance“ bekannt ist:

$$p_i w_{ij} = p_j w_{ji}. \quad (3.2)$$

Die Übergangswahrscheinlichkeiten und damit die Algorithmen sind mit diesen Bedingungen nicht eindeutig festgelegt. Für die Erzeugung von  $SU(3)$  Eichfeldern stehen im Wesentlichen zwei Algorithmen für die Generierung derartiger Markov-Ketten zur Verfügung: das Pseudowärmebad- [45] und das Metropolis-Verfahren [46]. Beide Verfahren können mit einer Aktualisierung der Eichfelder mittels Überrelaxation kombiniert werden.

Für das Metropolis-Verfahren muss für die Aktualisierung eines Eichfeldes  $U_{x,\mu}$  in geeigneter Weise ein Änderungsvorschlag generiert werden. Dies kann erreicht werden, indem die ursprüngliche  $SU(3)$  Matrix mit einer zufallsgenerierten  $SU(3)$  Matrix multipliziert wird. In unseren Routinen haben wir dazu eine  $3 \times 3$ -Matrix  $M$  erzeugt, so dass

$$M = R + bI, \quad (3.3)$$

wobei  $R$  eine Zufallsmatrix, deren Elemente im Intervall  $[-0.5, 0.5]$  gleichverteilt gewählt wurden, und  $I$  die Einheitsmatrix ist. Diese Matrix wird nun auf ein Gruppenelement der  $SU(3)$ -Gruppe projiziert und mit einer Wahrscheinlichkeit 0.5 transponiert. Die Projektion kann durch eine Gram-Schmidt-Orthonormalisierung realisiert werden. Durch die Wahl des Parameters  $b$  wird die Änderung gesteuert. Kleine Werte für  $b$  führen zu kleinen Schritten im Phasenraum, d.h. die Änderung wird zwar mit hoher Wahrscheinlichkeit akzeptiert, jedoch werden aufeinander folgende Konfigurationen stark korreliert sein. Umgekehrt führen große Werte für  $b$  zu kleinen Akzeptanzraten und damit weniger Änderungen. Empirisch findet man, dass das Optimum bei einer Akzeptanzrate von ca. 30% liegt.

Einer Aktualisierung durch Überrelaxation liegt die Annahme zugrunde, dass es auf einfache Art möglich ist, ein Gruppenelement zu bestimmen, welches die Wirkung (lokal) näherungsweise minimiert. Gelingt es nun, eine geeignete Transformation zu finden, die durch „Spiegelung“ an diesem Minimum zu einem Gruppenelement derart führt, dass die Änderung der Wirkung klein ist oder sogar verschwindet, dann ist es möglich, trotz großer Akzeptanzraten innerhalb des Phasenraums einen größeren Schritt zu machen.

Für Wirkungen der in Gleichung (2.8) gegebenen Form hat Creutz [47] einen Algorithmus vorgeschlagen, mit dessen Hilfe für  $U(1)$  neue Gruppenelemente gefunden werden können, die die Wirkung unverändert lassen. Für  $SU(3)$  ist die Änderung nicht Null, aber dennoch klein genug, um hohe Akzeptanzraten zu bekommen. Wir fanden in unseren Simulationen eine Rate von ca. 60%.

Für eine Aktualisierung aller Eichfelder auf dem Gitter haben wir diese in folgender Abfolge von Einzelschritten („sweeps“) aktualisiert:

```

for i = 1, ..., nsweep
  for j = 1, ..., nlink
    for k = 1, ..., nhit
      Do Metropolis update of link j
    end for
  end for
  for l = 1, ..., nover
    for j = 1, ..., nlink
      Do overrelaxation update of link j
    end for
  end for
end for

```

Bei unseren Simulationen haben wir `nsweep = 50`, `nhit=3` und `nover=16` gewählt.

Abbildung 3.1 zeigt den über eine Gitterkonfiguration gemittelten Erwartungswert für die Wilson-Plaquette

$$W_{11} = \frac{1}{18V} \sum_{\mu < \nu} \sum_x \text{Re tr } U_{x,\mu\nu}^{\square} \quad (3.4)$$

in Abhängigkeit von der Zahl der „sweeps“ unmittelbar nach einem Kaltstart. Ebenfalls eingezeichnet sind die nach Thermalisierung erreichten Werte, wie sie in Tabelle 3.1 aufgeführt sind. Eine Thermalisierung ist somit innerhalb einer Aktualisierung der Eichfelder erreicht.

Aus der normierten Autokorrelationszeit

$$\rho(i) = \frac{\langle W_{11}(i)W_{11}(0) \rangle - \langle W_{11}(i) \rangle \langle W_{11}(0) \rangle}{\langle W_{11}(0)W_{11}(0) \rangle - \langle W_{11}(0) \rangle \langle W_{11}(0) \rangle} \quad (3.5)$$

lässt sich die integrierte Autokorrelationszeit  $\tau_{int} = 1/2 + \sum_{i=1} \rho(i)$  berechnen. Mit  $i$  wird hier die Zahl der „sweeps“ bezeichnet. Wenn wir Madras und Sokal [48] folgend die Summation bei  $i = 6\tau_{int}$  abbrechen, erhalten wir für den mittleren Plaquettewert  $\tau_{int} \simeq 1.2 - 1.7$ . Daraus können wir schließen, dass die nach jedem zweiten „sweep“ ermittelten Plaquettewerte bereits statistisch unabhängig sind. Dies schließt jedoch nicht aus, dass weniger lokale Observablen, beispielsweise die in Gleichung (2.34) definierten Korrelationsfunktionen, stärker korreliert sind. Jedoch wurden diese nur nach jedem 50. „sweep“ bestimmt. Für diesen Fall konnten wir keine statistischen Korrelationen feststellen.

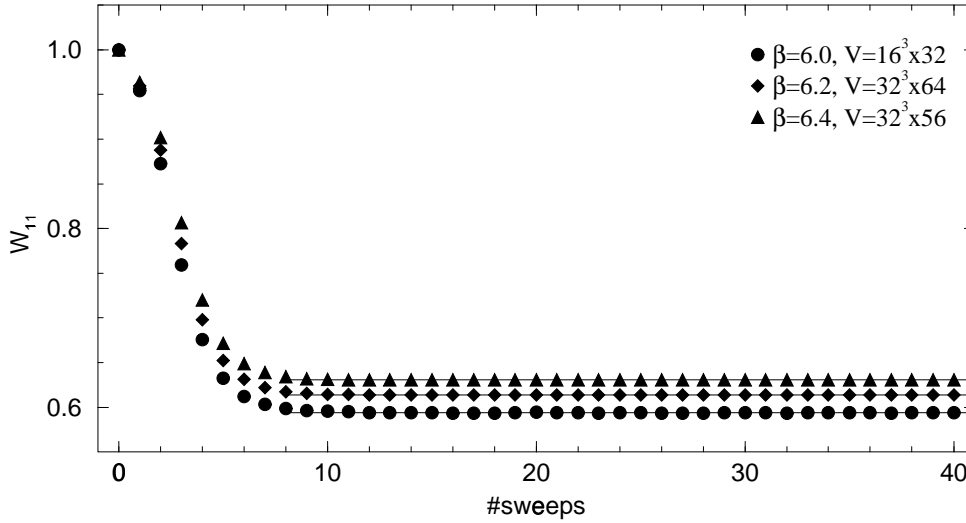


Abbildung 3.1.: Thermalisierung der Eichfeldkonfigurationen. Die durchgezogenen Linien zeigen die in Tabelle 3.1 aufgelisteten Mittelwerte für  $W_{11}$ .

| $\beta$ | Gitter           | Plaquette   |
|---------|------------------|-------------|
| 6.0     | $16^3 \times 32$ | 0.593679(8) |
| 6.2     | $32^3 \times 64$ | 0.613633(2) |
| 6.4     | $32^3 \times 64$ | 0.630633(4) |

Tabelle 3.1.: Erwartungswerte für die Wilson-Plaquette  $W_{11}$ .

## 3.2. Erzeugung von Propagatoren

Zur Berechnung von Quarkpropagatoren  $G_{xy}$  auf einer vorgegebenen Eichfeldkonfiguration müssen wir die Gleichung

$$\sum_y M_{x,y}^{\text{SW}} G_{y,x_0} = S_{x,x_0} \quad (3.6)$$

numerisch lösen. Wir werden im Folgenden zuerst die verwendeten Lösungsalgorithmen diskutieren, ehe wir auf die Wahl der Quelle  $S_{x,x_0}$  zurückkommen.

### 3.2.1. Inversionsalgorithmus

Die Fermionmatrix  $M_{x,y}^{\text{SW}}$  hat die Dimension  $3 \times 4 \times V$ . Bei einem Gitter der Größe  $32^3 \times 64$  umfasst diese Matrix  $25165824 \times 25165824$  Einträge, von denen jedoch aufgrund der einfachen Diskretisierung die meisten Null sind. Dennoch

bleibt die Lösung dieser Gleichung eine Herausforderung an Rechnertechnik und Algorithmen.

Bei der Lösung des mit Gleichung (3.6) definierten Problems sind zwei Eigenschaften der Matrix  $M_{x,y}^{SW}$  relevant: sie ist einerseits groß aber sehr dünn besetzt („*sparse*“) und andererseits nicht hermitesch.

Zur Lösung von Problemen der Form

$$Ax = b, \tag{3.7}$$

wobei  $A$  ein große aber dünn besetzte Matrix und  $x$  ein Lösungsvektor ist, werden in der Regel iterative Verfahren eingesetzt, bei denen aus einem anfänglichen Residuumsvektor  $r_0$  ein Krylov-Raum  $\mathcal{K}(A, r_0, k) = \text{span}\{r_0, Ar_0, \dots, A^{k-1}r_0\}$  konstruiert wird. Innerhalb dieses Raums werden orthogonale Suchvektoren  $p_k$  derart gewählt, dass  $\frac{1}{2}x^T Ax - x^T b$  minimiert wird. Bei diesen Algorithmen werden in unterschiedlicher Form folgende Schritte durchlaufen

```

initialize solution vector  $x_0$ 
 $r_0 = b - Ax_0$ 
 $k = 0$ 
while  $\|r_k\|/\|x_k\| > \epsilon$ 
     $k = k + 1$ 
    compute search vector  $p_k$ 
    compute  $\alpha_k \in \mathbb{C}$ , e.g.  $\alpha_k = p_k^T r_{k-1} / p_k^T A p_k$ 
     $x_k = x_{k-1} + \alpha_k p_k$ 
     $r_k = b - \alpha_k A p_k$ 
end while

```

Dem Problem fehlender Hermitezität kann dadurch begegnet werden, dass man statt Gleichung (3.7) die Gleichung

$$A^\dagger A x = A^\dagger b \tag{3.8}$$

löst. Allerdings bedeutet dies, dass sich die Zahl der Operationen „Matrix mal Vektor“ pro Iteration verdoppelt. Eine Alternative ist der Einsatz von Algorithmen, die die Hermitezität der Matrix  $A$  nicht voraussetzen. Dazu gehören der MR- („minimal residue algorithm“) [49] sowie der BiCGstab-Algorithmus („stabilized biconjugate gradient“) [50].

Der BiCGstab-Algorithmus konvergiert insbesondere für kleine Quarkmassen deutlich besser als der MR-Algorithmus [51]. Allerdings können Rundungsfehler dazu führen, dass die Konvergenz dieses Algorithmus sich verlangsamt oder dieser sogar fehlschlägt [52]. Um negative Effekte aufgrund von Rundungsfehlern zu vermeiden, haben wir den BiCGstab-Algorithmus alle 100-300 Iterationen durch Neuberechnung des Residuumsvektors  $r_k$  neu gestartet.

Den MR-Algorithmus haben wir eingesetzt, wenn der BiCGstab-Algorithmus nicht innerhalb einer vorgegebenen Anzahl von Iterationsschritten konvergierte. Als Konvergenzkriterium haben wir die Bedingung

$$\frac{\|r_k\|}{\|x_k\|} < \epsilon \quad (3.9)$$

verwendet. Die Rechnungen wurden auf den Rechnern vom Typ APE/Quadratics architekturbedingt grundsätzlich in einfacher Genauigkeit mit  $\epsilon = 10^{-6}$  durchgeführt. Die Simulationen bei  $\beta = 6.2$  auf Gittern der Größe  $32^3 \times 64$  wurden in doppelter Genauigkeit auf Rechnern vom Typ Cray/T3E realisiert, wobei  $\epsilon = 10^{-15}$  gewählt wurde. Bei weiteren Simulationen auf Rechnern dieses Typs bei größeren Quarkmassen wurden für Vektoren einfache, für skalare Objekte doppelt genaue Gleitkommazahlen verwendet. Hier haben wir überwiegend  $\epsilon = 10^{-6}$  gesetzt. Ein Vergleich mit Simulationen, bei denen  $\epsilon = 10^{-8}$  gesetzt worden war, ergaben keine Hinweise auf systematische Unterschiede.

Für Simulationen mit einer modifizierten Wirkung (siehe Abschnitt 4.8) haben wir ferner den „conjugate gradient“-Algorithmus (CG) [53] eingesetzt. Dieser Algorithmus setzt jedoch eine hermitesche Matrix voraus.

### 3.2.2. Präkonditionierung

Für die numerische Lösung eines Gleichungssystems der in (3.7) gegebenen Form kann es vorteilhaft sein, stattdessen das Gleichungssystem  $\tilde{A} \tilde{x} = \tilde{b}$  zu lösen, wobei  $\tilde{A} = V_1^{-1} M V_2^{-1}$ ,  $\tilde{x} = V_2 x$  und  $\tilde{b} = V_1^{-1} b$  ist. Dies ist immer dann der Fall, wenn die Matrix  $\tilde{A}$  besser konditioniert ist (und sich somit leichter invertieren lässt) und außerdem die Bestimmung von  $\tilde{b}$  sowie die Lösung der Gleichung  $\tilde{x} = V_2 x$  nicht aufwendig ist.

Die Fermionmatrix, so wie sie in Gleichung (2.22) definiert ist, zerfällt auch nach Ergänzung des Verbesserungsterms der Sheikholeslami-Wohlert-Wirkung in zwei Teile: einen im Ortsraum diagonalen Teil und einen Teil, der jeweils zwei benachbarte Gitterpunkte miteinander verknüpft. Die Gitterpunkte lassen sich nun in „gerade“ (even) oder „ungerade“ (odd) unterscheiden, je nachdem, ob sich die jeweiligen Koordinaten  $(x_1, x_2, x_3, x_4)$  zu einer geraden oder ungeraden Zahl addieren. Nummeriert man die Gitterplätze so, dass zuerst alle Nummern an ungerade und danach an gerade Plätze vergeben werden, so lässt sich die Fermionmatrix wie folgt schreiben:

$$M^{SW} = \begin{pmatrix} X & \kappa D_{oe} \\ \kappa D_{eo} & X \end{pmatrix}, \quad (3.10)$$

wobei

$$X = \left[ 1 + \frac{i\kappa c_{sw}}{2} \sigma_{\mu\nu} F_{\mu\nu} \right] \delta_{x,y} \quad (3.11)$$

im Ortsraum diagonal ist. Verwenden wir

$$V_1 = \begin{pmatrix} X & 0 \\ \kappa D_{eo} & X \end{pmatrix} \quad \text{und} \quad V_2 = \begin{pmatrix} X & \kappa D_{oe} \\ 0 & X \end{pmatrix}, \quad (3.12)$$

so ergibt sich

$$\tilde{A} = \begin{pmatrix} X^{-1} & 0 \\ 0 & X^{-1}(1 - \kappa^2 D_{eo} X^{-1} D_{oe} X^{-1}) \end{pmatrix}. \quad (3.13)$$

Es ist somit ausreichend, zuerst nur die geraden Gitterplätze zu betrachten und die Gleichung

$$(X - \kappa^2 D_{eo} X^{-1} D_{oe}) x_e = b_e - \kappa D_{eo} X^{-1} b_o \quad (3.14)$$

zu lösen. Der Lösungsvektor  $\tilde{x}$  muss gar nicht explizit berechnet werden. Bei der Matrix auf der linken Seite von Gleichung (3.14) sind die im Ortsraum nicht-diagonalen Elemente proportional  $\kappa^2$  und, da  $\kappa < 1$  ist, kleiner als bei der ursprünglichen Matrix  $A$ . Ist der gerade Teil des Lösungsvektors  $x_e$  bekannt, so ist der ungerade Teil gegeben durch

$$x_o = X^{-1} (b_o - \kappa D_{oe} x_e). \quad (3.15)$$

Durch diese Art der Präkonditionierung, die als „even-odd-preconditioning“ bekannt ist, wird ein Faktor 2-3 an Rechenzeit gewonnen [54].

### 3.2.3. Wahl der Quellen

Die Quelle  $S_{x,x_0}$  ist so zu wählen, dass die berechneten Korrelationsfunktionen ein möglichst gutes Signal liefern. Die einfachste Wahl ist

$$S_{x,x_0} = \delta_{x,x_0}. \quad (3.16)$$

Physikalisch wird man jedoch erwarten, dass eine Quelle mit einer dem zu untersuchenden Teilchen entsprechenden Ausdehnung ein wesentlich besseres Signal liefert. Bei der Verwendung ausgedehnter Quellen müssen entweder diese eichinvariant gewählt oder die Eichfeldkonfigurationen eichfixiert werden.

Die Quelle in Gleichung (3.16) kann [55] durch Lösung einer dreidimensionalen Klein-Gordon-Gleichung

$$\sum_{\vec{x}'} K_{\vec{x},\vec{x}'} S_{\vec{x}',\vec{x}_0} = \delta_{\vec{x},\vec{x}_0} \quad (3.17)$$

in der Zeitschicht  $t_0$  eichinvariant zu einer ausgedehnten Quelle verschmiert werden, wobei  $K_{\vec{x},\vec{x}'}$  durch

$$K_{\vec{x},\vec{x}'} = \delta_{\vec{x},\vec{x}'} - \kappa_S \sum_i \left[ \delta_{\vec{x}',\vec{x}-i} U_{\vec{x}-i,i}^\dagger + \delta_{\vec{x}',\vec{x}+i} U_{\vec{x},i} \right] \quad (3.18)$$

gegeben ist. Dieses Verfahren wird „Wuppertal-Verschmierung“ genannt. Es lässt sich vereinfachen [56], indem Gleichung (3.17) mit Hilfe von  $N$  Jacobi-Iterationen nach Potenzen von  $\kappa_S$  entwickelt wird. Dieses Verfahren, welches auch im Rahmen der vorliegenden Arbeit angewandt wurde, ist als „Jacobi-Verschmierung“ bekannt.

Mit beiden Verfahren können Quellen erzeugt werden, die im Mittel symmetrisch über einen Radius  $r$  um den Punkt  $x_0$  verschmiert sind, wobei  $r$  gegeben ist durch

$$r^2 = \frac{\sum_{\vec{x}} |\vec{x}|^2 |S_{\vec{x}\vec{x}_0}|^2}{\sum_{\vec{x}} |S_{\vec{x}\vec{x}_0}|^2}. \quad (3.19)$$

$r$  ist bei der Anwendung der Jacobi-Verschmierung eine Funktion von  $\kappa_S$  und  $N$ . Die von uns verwendeten Parameter sind in Tabelle 3.2 aufgelistet. Als Position der Quelle haben wir ausschließlich den Punkt  $x_0 = (\vec{0}, 0)$  verwendet.

| $\beta$ | $\kappa_S$ | $N$ | $r/a$  | $r$ [fm] |
|---------|------------|-----|--------|----------|
| 6.0     | 0.21       | 50  | 3.8(1) | 0.35(1)  |
| 6.2     | 0.21       | 100 | 5.5(2) | 0.37(1)  |
| 6.4     | 0.21       | 150 | 6.7(2) | 0.34(1)  |

Tabelle 3.2.: Bei der Jacobi-Verschmierung verwendete Parameter und die von uns bestimmten mittleren Radien der Quellen. Angegeben ist außerdem die Standardabweichung von  $r$ .

### 3.2.4. Verbesserung des Signals

Das Signal kann weiter verbessert werden, indem der Algorithmus des Verschmierens nicht nur auf die Quelle bei  $x_0 = (\vec{x}_0, t_0)$  sondern auch auf den Propagator  $G_{x,x_0}$  bei  $x = (\vec{x}, t)$  angewendet wird. Der Effekt ist in Abbildung 3.2 schematisch dargestellt.

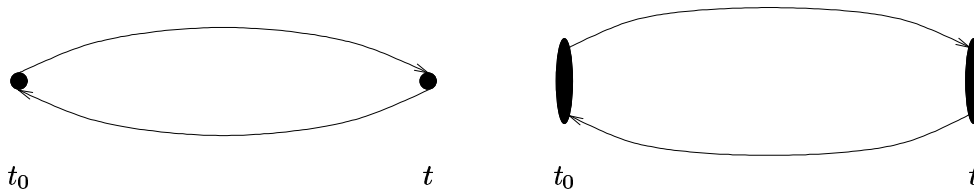


Abbildung 3.2.: Schematische Darstellung einer unverschmierten (links) und einer verschmierten (rechts) mesonischen Korrelationsfunktion.



Eine weitergehende Unterdrückung angeregter Zustände kann durch Anwendung einer nicht-relativistischen Projektion erfolgen [57], bei der Quarkfelder wie folgt ersetzt werden:

$$\begin{aligned}\psi &\rightarrow \psi^{\text{NR}} = \frac{1}{2}(1 + \gamma_4) \psi \\ \bar{\psi} &\rightarrow \bar{\psi}^{\text{NR}} = \bar{\psi} \frac{1}{2}(1 + \gamma_4).\end{aligned}\quad (3.20)$$

Durch die nicht-relativistische Projektion kann zwar erreicht werden, dass bereits für kleine  $t$  der Grundzustand dominiert, jedoch werden unter Umständen zu viele Informationen verworfen. Dadurch kann effektiv der statistische Fehler durch Anwendung dieser Projektion größer werden.

Für die in Gleichung (2.34) definierten Korrelationsfunktionen erwarten wir ein zeitlich exponentiell abfallendes Verhalten, wenn der zeitliche Abstand zur Quelle groß ist. Für die sogenannte *effektive Masse*

$$am_{\text{eff},t} = \ln \left| \frac{C_t}{C_{t+1}} \right| \quad (3.21)$$

erwarten wir ein konstantes Verhalten, sobald der Grundzustand der dominierende Zustand ist. Abbildung 3.3 zeigt die effektive Masse des Vektormesons

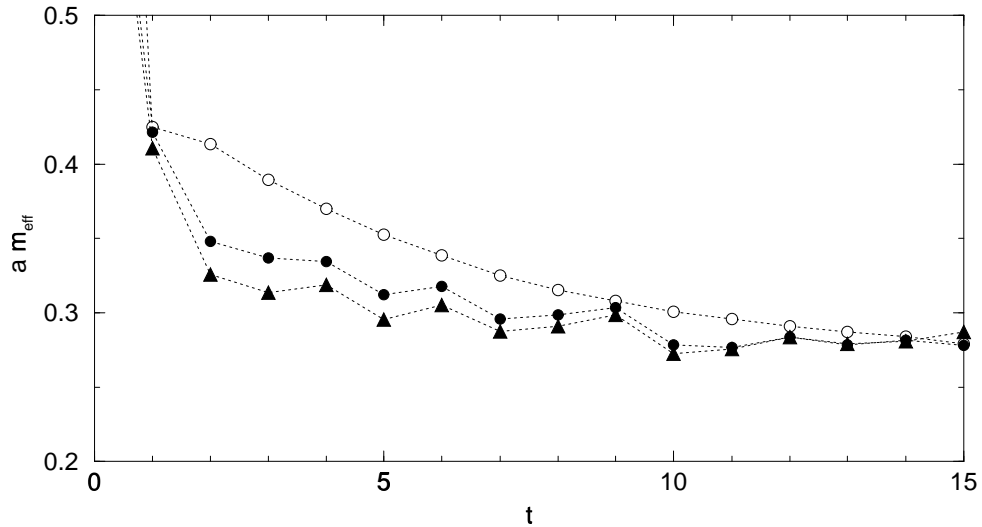


Abbildung 3.3.: Effektive Masse  $m_{\text{eff}}$  des Vektormesons für  $\beta = 6.4$ ,  $\kappa = 0.1350$  und  $V = 32^3 \times 48$ . Bei Verschmieren von Quelle und Propagator (●) erreicht  $m_{\text{eff}}$  früher einen konstanten Wert als ohne Verschmieren des Propagators (○). Die Anwendung der nicht-relativistischen Projektion (▲) verstärkt diesen Effekt nur leicht. Die Fehlerbalken sind hier nicht dargestellt.

ohne und mit Anwendung der soeben beschriebenen Verfahren zur Verbesserung des Signals.

Durch Anwendung verschmierter Quellen und Propagatoren bekommen wir auch für das pseudoskalare Meson und das Nukleon ein gutes Signal. In Abbildung 3.4 sind die effektiven Massen im Vergleich dargestellt. Für die Verschmierung der Propagatoren haben wir die gleichen Parameter verwendet wie für die Verschmierung der Quellen.

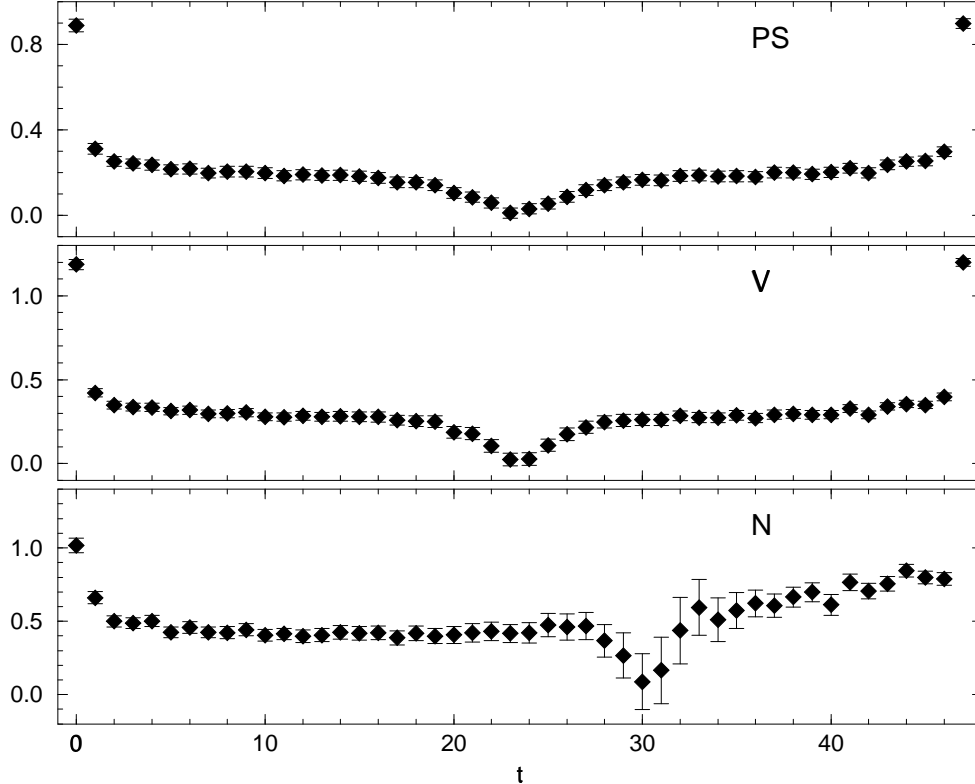


Abbildung 3.4.: Betrag der effektiven Masse  $|m_{\text{eff}}|$  des pseudoskalaren (oben) und des Vektormesons (Mitte) sowie des Nukleons (unten) für  $\beta = 6.4$ ,  $\kappa = 0.1350$  und  $V = 32^3 \times 48$ . In allen drei Fällen wurden sowohl Quellen wie Propagatoren verschmiert. Im Falle des Nukleons wurde außerdem die nicht-relativistische Projektion angewendet.

### 3.3. Wahl der Simulationsparameter

Unsere Simulationen wurden für drei verschiedene Werte des Eichfeldkoppelungsparameters  $\beta$  durchgeführt. Dies entspricht drei physikalischen Werten des Gitterabstandes  $a$ . Rechnungen für schwere Quarkmassen wurden auf Gittern mit einer räumlichen Ausdehnung von etwa 1.6 fm durchgeführt. Bei den

leichten Quarkmassen war das Gitter 2.2 fm groß. Die einzelnen Werte für  $\beta$  und die Gittergrößen sind in Tabelle 3.3 zusammengestellt.

Für jeden Wert von  $\beta$  wurden die hadronischen Observablen für acht bis elf verschiedene Quarkmassen bestimmt. Die Masse des pseudoskalaren Mesons  $m_{\text{PS}}$  lag dabei zwischen 0.4 und 3.9 GeV. Das Verhältnis von  $m_{\text{PS}}$  zur Masse des Vektormesons  $m_{\text{V}}$  variierte von 0.41 bis 0.97. Die verwendeten Werte für den Kopplungsparameter  $\kappa$  sind im Anhang A.1 aufgelistet.

| $\beta$ | $a$ [fm] | Gitter              | Räuml.Ausdehnung [fm] |
|---------|----------|---------------------|-----------------------|
| 6.0     | 0.093    | $16^3 \times 32$    | 1.5                   |
|         |          | $24^3 \times 32$    | 2.2                   |
| 6.2     | 0.068    | $24^3 \times 48$    | 1.6                   |
|         |          | $32^3 \times 64$    | 2.2                   |
| 6.4     | 0.051    | $32^3 \times 48/64$ | 1.6                   |

Tabelle 3.3.: Bei den Simulationen verwendete Werte der Eichfeldkopplungen  $\beta$  und Gittergrößen. Die Umrechnung in physikalische Einheiten erfolgte mit der in Abschnitt 4.5 definierten  $r_0$ -Skala.

### 3.4. Datenanpassung und Fehlerbestimmung

Für die Bestimmung hadronischer Observablen ist es notwendig, verschiedene Ansätze an unsere Ergebnisse anzupassen. Dies geschieht mit Hilfe einer  $\chi^2$ -Minimierung, wobei  $\chi^2$  ein Maß für die Abweichung der Daten vom Fitansatz  $f_{a_1 \dots a_M, t}$  ist mit  $a_1 \dots a_M$  den zu bestimmenden Fitparametern. Im Idealfall Gauß-verteilter und statistisch unabhängiger Ergebnisse erwarten wir, dass  $\chi^2$  etwa gleich der Zahl der Freiheitsgrade und somit  $\chi^2_{\text{pdf}} \simeq 1$  ist.

Bei der Berechnung von Korrelationsfunktionen auf dem Gitter können wir für jede Zeitschicht  $t$  mit Hilfe der Ergebnisse  $y_{i,t}$  für  $N$  Konfigurationen  $i$  einen Mittelwert  $\mu_t$  und eine Standardabweichung des Mittelwertes  $\sigma_t$  bestimmen.  $\chi^2$  wird dann in der Regel wie folgt definiert:

$$\chi^2_{a_1 \dots a_M} = \sum_t \frac{(f_{a_1 \dots a_M, t} - \mu_t)^2}{\sigma_t^2}. \quad (3.22)$$

Diese Definition berücksichtigt allerdings nicht, dass unsere Ergebnisse zwar für verschiedene Konfigurationen  $i$  unkorreliert, für verschiedene Zeitschichten  $t$  jedoch stark korreliert sind. Im Allgemeinen ist daher zu erwarten,

### 3. Realisierung der Simulationen

---

dass wir dadurch  $\chi^2$  unterschätzen. Führen wir nun die Kovarianzmatrix

$$c_{t,t'} = \frac{1}{N(N-1)} \sum_{i=1}^N (y_{i,t} - \mu_t)(y_{i,t'} - \mu_{t'}) \quad (3.23)$$

ein, so gelangen wir zu der Definition

$$\chi_{a_1 \dots a_M}^2 = \sum_{t,t'} \frac{(f_{a_1 \dots a_M,t} - \mu_t)(f_{a_1 \dots a_M,t'} - \mu_{t'})}{c_{t,t'}}. \quad (3.24)$$

Sind die Daten tatsächlich unabhängig, so ist  $c_{t,t'} = \sigma_t^2 \delta_{t,t'}$ . Dies führt dann zurück auf die ursprüngliche Definition.

Zwar erlaubt die Berücksichtigung der Kovarianzmatrix prinzipiell eine zuverlässigere Abschätzung von  $\chi^2$ . Allerdings können kleine Eigenwerte dieser Matrix dazu führen, dass die Berechnung ihres Inversen unzuverlässig wird.<sup>1</sup> Dies verändert nicht nur die Abschätzung von  $\chi^2$ , sondern kann auch falsche Ergebnisse für die Fitparameter  $a_1 \dots a_M$  zur Konsequenz haben. Um einer solchen Situation zu begegnen, haben wir zur Berechnung der Inversen der Kovarianzmatrix die Singulärwertzerlegung („*singular value decomposition*“, SVD) eingesetzt [59].

Das soeben beschriebene Problem tritt immer dann auf, wenn die vorhandene Zahl von Konfigurationen klein ist. Wieviele Konfigurationen erforderlich sind, ist abhängig von der betrachteten Observable oder, anders ausgedrückt, vom Verhältnis von Signal zu Rauschen. Fits an die Ergebnisse unserer Simulationen haben wir grundsätzlich sowohl mit der diagonalen Kovarianzmatrix als auch der vollständigen Kovarianzmatrix durchgeführt. Abweichungen zwischen den Ergebnissen betrachten wir als ein Maß für die Qualität unserer Daten. Aufgrund der größeren Stabilität der Resultate, die durch Fits mit der diagonalen Kovarianzmatrix bestimmt wurden, haben wir nur diese bei der weiteren Analyse verwendet.

Das Verhältnis von Signal zu Rauschen wird in der Regel umso schlechter, je weiter wir von der Quelle entfernt sind. In der Nähe der Quelle müssen wir jedoch davon ausgehen, dass das Verhalten der Korrelationsfunktionen noch von angeregten Zuständen bestimmt ist. Um das Fitintervall geeignet zu wählen, haben wir dieses systematisch so lange verkleinert, bis die Fitparameter  $a_i$  innerhalb der Fehler unabhängig vom Fitintervall waren. Das Fitintervall haben wir dabei in der Regel symmetrisch gewählt.

Da die Werte für die Fitparameter  $a_i$  das Ergebnis von Fits an korrelierten Daten sind, ist die übliche Bestimmung des Fehlers für diese Parameter mit Hilfe von  $\chi^2$  unzuverlässig. Zur Fehlerbestimmung haben wir daher die Bootstrap-Methode gewählt [60]. Dabei wird mit Hilfe eines Satzes von  $N$

<sup>1</sup> Für eine weitergehende Diskussion dieses Problems siehe [58].

ganzahligen Zufallszahlen  $k$ , die jeweils mit gleicher Wahrscheinlichkeit den Wert  $1, 2, \dots, N$  annehmen, ein neues Ensemble von Ergebnissen erzeugt, auf welches anschließend die gleichen Fitprozeduren wie auf das ursprüngliche Ensemble angewendet werden. Wiederholt man diese Prozedur, so erhält man schließlich eine statistische Verteilung für die Fitparameter  $a_i$ . Dies erlaubt die Bestimmung einer Standardabweichung für jeden einzelnen Parameter.

Wir haben in dieser Arbeit in der Regel 200 derartiger Bootstrap-Ensembles generiert. Als Ergebnis für einen Fitparameter  $a_i$  werden wir das Resultat eines Fits an die ursprünglichen Daten angeben. Sofern nicht anders angegeben, gibt der jeweils zitierte Fehler die durch die Bootstrap-Methode bestimmte Standardabweichung an.

### 3. Realisierung der Simulationen

---