





Institut für Theoretische Physik

## Bachelorarbeit

im Studiengang Physik

zur Erlangung des akademischen Grades  
Bachelor of Science

**Thema:** Bestimmung von Hadronmassen aus Korrelatoren  
durch Fitten von Exponentialfunktionen

**Autor:** Lutz Kiefer  
MatNr. 4977074

**Version vom:** 22. September 2015

**1. Betreuer:** Prof. Dr. Marc Wagner  
**2. Betreuer:** Prof. Dr. Owe Phillipsen

## Zusammenfassung

In meiner Arbeit lege ich dar, dass sich aus Korrelationsfunktionen Massen von Hadronzuständen gewinnen lassen. Ich werde dabei nachweisen, dass man dies durch das Fitten von Exponentialfunktionen an die Korrelatoren erreichen kann. Das Hauptaugenmerk liegt dabei darauf, zu zeigen, dass die Ergebnisse dieser Methode die gleichen Ergebnisse liefert wie das Generalisierte-Eigenwert-Problem(GEP). Zusätzlich werde ich das Programm, das ich für diese Arbeit geschrieben habe, kurz vorstellen bevor ich zum Schluss noch kurz auf das D-Meson und das Charmonium als Beispiel zum Gewinnen von Ergebnissen eingehe.

## Abstract

In my work I will show, that you can extract hadron masses from correlation functions. I will show aswell that you can get these masses by fitting exponentials to the correlators. The main task is to show, that these results are consistent with the results of the Generalized-Eigenvalue-Problem(GEP). Additionally I will introduce the functions of the program I wrote. At the end I will show the D-meson and the charmonium as example.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>5</b>
<b>Tabellenverzeichnis</b>	<b>5</b>
<b>1 Einleitung</b>	<b>6</b>
<b>2 Theoretische Grundlagen</b>	<b>7</b>
2.1 Quantenchromodynamik . . . . .	7
2.2 Pfadintegralformalismus . . . . .	7
2.3 Gitter QCD . . . . .	8
2.4 Wilson twisted mass Fermionen . . . . .	10
2.5 Klassifizierung von Hadronen und Erzeugungsoperatoren . . . . .	10
2.5.1 Klassifizierung von Hadronen . . . . .	10
2.5.2 Ladungskonjugation C . . . . .	11
2.5.3 Parität P . . . . .	11
2.5.4 Erzeugungsoperatoren für Mesonen . . . . .	11
2.5.5 Erzeugungsoperatoren für Baryonen . . . . .	12
2.6 Bestimmung von Massen . . . . .	13
2.6.1 Korrelatoren . . . . .	13
2.6.2 Korrelator-Matrizen . . . . .	14
2.7 $\chi^2$ minimierende Fits . . . . .	16
2.8 Binning . . . . .	16
2.9 Die delete-1-Jackknife-Methode . . . . .	17
<b>3 Programmierarbeit</b>	<b>17</b>
3.1 Anforderungen des Programmes . . . . .	18
3.2 Ablauf des Programms und Flussdiagramm . . . . .	18
3.3 Simultanes Fitten . . . . .	19
3.4 Das Optionsfile . . . . .	22
3.5 Ausgabe PDF des Programmes . . . . .	23
3.6 Probleme des Programms und mögliche Lösungen . . . . .	25
<b>4 Ergebnisse und Interpretation</b>	<b>27</b>
4.1 D-Meson $1 \times 1$ Matrix : $J = 0$ . . . . .	28
4.2 D-Meson $2 \times 2$ Matrix : $J = 0$ . . . . .	29
4.3 D-Meson $4 \times 4$ Matrix: $J = 0$ . . . . .	30
4.4 D-Meson $4 \times 4$ Matrix $J = 0, P = -$ . . . . .	32
4.5 Charmonium $6 \times 6$ Matrix: $J = 0$ . . . . .	34
<b>5 Ausblick</b>	<b>35</b>
<b>6 Fazit</b>	<b>35</b>
<b>Literaturverzeichnis</b>	<b>37</b>

<b>Danksagung</b>	<b>37</b>
<b>Eidesstattliche Erklärung</b>	<b>39</b>

## Abbildungsverzeichnis

1	Flussdiagramm . . . . .	19
2	Ausgabe des Programmes . . . . .	24
3	Minimierungsprobleme Beispiel . . . . .	25
4	Tabelle der $\Gamma$ s [KW15] . . . . .	28
5	$1 \times 1$ Matrizen . . . . .	29
6	$1 \times 1$ Matrizen . . . . .	30
7	$2 \times 2$ Matrix 2 Energien GEP . . . . .	30
8	$4 \times 4$ Matrizen . . . . .	31
9	$6 \times 6$ Matrizen . . . . .	34
10	$2 \times 2$ Matrix 2 Energien GEP . . . . .	34

## Tabellenverzeichnis

1	Beispiele $\Gamma$ [GL] . . . . .	12
2	Siehe [KW15, Kap. 4.1 Table 1] . . . . .	27

# 1 Einleitung

Im Laufe des 20. Jahrhunderts wandelte sich unser Bild vom Aufbau der Materie grundlegend. In den 1930er Jahren glaubte man, mit den Elementarteilchen Proton, Neutron und Elektron, dem Myon sowie den Neutrinos den Aufbau der Materie vollständig verstanden zu haben.

Mit dem technischen Fortschritt und der Entwicklung besserer Teilchenbeschleuniger konnte man jedoch immer mehr neue Teilchen und weitere angeregte Zustände entdecken. Wegen der großen Anzahl an neu entdeckten Hadronen führte man scherzhaft den Begriff "Teilchenzoo" ein. Jedoch lies die große Anzahl der neu entdeckten Hadronen Zweifel daran wachsen, ob es sich tatsächlich um Elementarteilchen handelt. Vielmehr vermutete man, dass es ein grundlegendes Prinzip für den Aufbau der Hadronen gibt, ähnlich dem Aufbau der chemischen Elemente, die schließlich durch unterschiedliche Kombinationen von Protonen im Kern des Atoms entstehen.

Ein solches grundlegendes Prinzip schlugen 1963 die Physiker Murray Gell-Mann und Georg Zweig vor [PS]. Gell-Mann postulierte, dass Hadronen gebundene Zustände aus sogenannten Quarks sind, wobei die Quarks die eigentlichen Elementarteilchen sind. Er erkannte, dass man die Hadronen mit Hilfe der symmetrischen unitären Gruppe  $SU(3)$  strukturieren kann. Diese Erkenntnis führte schließlich zur Entwicklung der Quantenchromodynamik(QCD), die heute als die Theorie gilt, die die starke Wechselwirkung beschreibt. Das Quarkmodell und die QCD sind heute allgemein anerkannt, da ihre Vorhersagen durch Experimente im Rahmen der Fehler bestätigt werden konnten.

Mit dem Quarkmodell lassen sich die Hadronen in zwei Gruppen teilen, die Baryonen und die Mesonen. Die Baryonen haben einen halbzahligen Spin und bestehen aus drei Quarks. Die Mesonen haben einen ganzzahligen Spin und bestehen aus einem Quark-Antiquark-Paar.

## 2 Theoretische Grundlagen

In diesem Kapitel werden die grundlegenden theoretischen Prinzipien der Gitter-QCD mit Blick auf ihre Relevanz für diese Arbeit vorgestellt. Für eine detailliertere Einführung verweise ich auf [Rot] und [GL].

### 2.1 Quantenchromodynamik

Die Quantenchromodynamik (QCD) ist die fundamentale Eichtheorie zur Beschreibung der starken Wechselwirkung. Sie basiert auf der nicht-abelschen Eichgruppe  $SU(3)$ .

Die zu beschreibenden Felder der QCD sind Quarkfelder. Dabei handelt es sich um fermionische Felder  $\Psi_f$ . Die Wechselwirkung zwischen den Quarkfeldern wird von den Eichfeldern  $G_\mu$  vermittelt.

Die Bewegungsgleichungen für die einzelnen Felder folgen aus dem Hamiltonschen-Prinzip. Die Grundlage dafür bildet die Lagrangedichte  $\mathcal{L}$

$$\mathcal{L}_{QCD}[\Psi, \bar{\Psi}, G_\mu] = \sum_f \bar{\Psi}_f (i\gamma_\mu \mathcal{D}_\mu + m_f) \Psi_f + \frac{1}{4} F_{\mu\nu}^c F_{\mu\nu}^c$$

$$\mathcal{D}_\mu = \partial_\mu - igG_\mu$$

aus der die Wirkung  $S[\Psi, \bar{\Psi}, G_\mu]$  per Integration über die gesamte Raumzeit folgt

$$S[\Psi, \bar{\Psi}, G_\mu] = \int d^4x \mathcal{L}_{QCD}[\Psi, \bar{\Psi}, G_\mu].$$

Für die Gluonfelder  $[G_\mu]$  und den Feldstärketensor  $[F_{\mu\nu}^c]$  des gluonischen Feldes gilt

$$G_\mu = G_\mu^c \frac{\lambda^a}{2}, a \in [1, \dots, 8]$$

$$F_{\mu\nu}^c = \partial_\mu G_\nu^c - \partial_\nu G_\mu^c + gf^{cde} G_\mu^d G_\nu^e.$$

Hierbei sind  $\lambda^a$  die Generatoren und  $f^{cde}$  die Strukturkonstanten der Lie-Algebra der  $SU(3)$ . Man nennt die Generatoren der  $SU(3)$  auch Gell-Mann-Matrizen.

### 2.2 Pfadintegralformalismus

Die in dieser Arbeit untersuchten Observablen werden durch Korrelationsfunktionen berechnet. Im allgemeinen haben Korrelationsfunktionen folgende Form:

$$\langle \Omega | \hat{O}_0(t_0) \dots \hat{O}_n(t_n) | \Omega \rangle, t_0 < t_1 < \dots < t_n$$

Korrelationsfunktionen können mit Hilfe von Funktionalintegralen folgendermaßen berechnet werden[PWL13]

$$\langle \Omega | \hat{O}_0(t_0) \dots \hat{O}_n(t_n) | \Omega \rangle = \frac{1}{Z} \int \mathcal{D}\Psi \mathcal{D}\bar{\Psi} \mathcal{D}A_\mu O_0 \dots O_n \exp(-S_E[\Psi, \bar{\Psi}, A_\mu])$$

$$Z = \int \mathcal{D}\Psi \mathcal{D}\bar{\Psi} \mathcal{D}A_\mu \exp(-S_E[\Psi, \bar{\Psi}, A_\mu]).$$

Beachte, dass rechts des Gleichheitszeichen die Operatoren  $\hat{O}_j$  durch Funktionale  $O_j$  ersetzt werden.

Wichtig ist dabei zu beachten, dass  $S_E[\Psi, \bar{\Psi}, A_\mu]$  die euklidische Wirkung der QCD ist, die aus dem Übergang  $t \rightarrow -i\tau$  generiert wird. Das Integrationsmaß des Pfadintegrals ist definiert durch:

$$\mathcal{D}\Psi \mathcal{D}\bar{\Psi} \mathcal{D}A_\mu := \prod_{x,\alpha} \prod_{y,\beta} \prod_{z,\mu} \mathcal{D}\Psi_\alpha(x) \mathcal{D}\bar{\Psi}_\beta(y) \mathcal{D}A_\mu(z)$$

Im Falle der QCD muss außerdem beachtet werden, dass es sich bei  $\bar{\Psi}$  und  $\Psi$  um Graßmannvariablen handelt. Genauere Informationen zu Graßmannvariablen findet man bei [BGHZ].

Im Allgemeinen lassen sich solche Pfadintegrale nicht analytisch lösen. Jedoch bieten sie einen numerischen Zugang zur Berechnung von Vakuumerwartungswerten (VEVs).

## 2.3 Gitter QCD

Da sich Pfadintegrale im Allgemeinen nicht analytisch lösen lassen, muss man auf numerische Methoden ausweichen. Damit dies möglich ist, geht man von der kontinuierlichen in eine diskretisierte Raumzeit über, indem man ein Gitter mit konstantem Gitterabstand  $a$  definiert. Diesen Schritt nennt man Regularisierung. Dabei gilt es zu beachten, dass die Eigenschaften der QCD wie z.B. die Eichinvarianz erhalten bleiben.

Durch die Gitterregularisierung entstehen folgende Neudefinitionen [[GL],[Rot]:

1.  $x_\mu \rightarrow an_\mu$
2.  $\Phi(x) \rightarrow \Phi(na)$
3.  $\int d^x \rightarrow a^4 \sum_n$
4.  $\mathcal{D}\Phi \rightarrow \prod_n d\Phi(na)$

Auch die Ableitungen müssen in eine für das Gitter geeignete Form gebracht werden. Hierfür nutzt man den Differenzenquotient. Dabei ist darauf zu achten, dass es verschiedene Möglichkeiten für den Differenzenquotient und damit verschiedene Regularisierungsmöglichkeiten gibt. Diese sind jedoch im Kontinuumslimit äquivalent, während der Gitterrechnungen muss man allerdings konsequent denselben nutzen um konsistente Ergebnisse



zu erhalten. Im folgenden werde ich den Vorwärts- und Rückwärts-Differenzenquotienten nutzen:

$$5. \partial_\mu F(x) \rightarrow \Delta_\mu^f = \frac{F(x+a\hat{\mu})-F(x)}{a}$$

$$6. \partial_\mu F(x) \rightarrow \Delta_\mu^b = \frac{F(x)-F(x-a\hat{\mu})}{a}$$

Wendet man diese Form jedoch auf Eichtheorien an, stellt sich heraus, dass die Eichinvarianz gebrochen ist. Damit die Eichinvarianz erhalten bleibt, führt man analog zur kontinuierlichen Theorie die kovariante Gitterableitung ein:

$$D_\mu^f \Phi(x) = \frac{U(x, x+a\hat{\mu})\Phi(x+a\hat{\mu}) - \Phi(x)}{a} \quad (1)$$

$$D_\mu^b \Phi(x) = -\frac{U(x, x+a\hat{\mu})\Phi(x+a\hat{\mu}) - \Phi(x)}{a} \quad (2)$$

$U(x,y)$  ist ein sogenannter Paralleltransporter, auch Link genannt. Er ist definiert durch:

$$U(x, y) = \exp \left( \int_x^y dz_\mu A_\mu(z) \right)$$

und besitzt folgendes Transformationsverhalten [GL],[Rot]:

$$U(x, y) \rightarrow U'(x, y) = g(x)U(x, y)g^\dagger(y)$$

wodurch die Eichinvarianz wieder hergestellt wird. Nutzen wir nun die bisher angesprochenen Möglichkeiten zur Gitterregularisierung ergibt sich beispielsweise folgende Gitterwirkung für Fermionen:

$$S_f[\Psi, \bar{\Psi}, U] = a^4 \sum_n \bar{\Psi}_f(n) \left( \frac{1}{2} \gamma_\mu (D_\mu^f + D_\mu^b) + m \right) \Psi_f(n)$$

Nutzt man diese naive Wirkung, tritt das sogenannte "fermion doubling problem" auf. Weitere Details zum "fermion doubling problem" findet man bei [Rot, Kap. 4.1 und 4.2]. Die Gitterwirkung für Eichfelder ergibt sich zu [Rot]:

$$S_G[U] = \frac{1}{g^2} \sum_n \left( 1 - \frac{1}{2} (P_{\mu\nu} + P_{\mu\nu}^\dagger) \right)$$

mit der sogenannten Plakette  $[P_{\mu\nu}]$ , gegeben durch:

$$P_{\mu\nu} = Tr \left( U_\mu(n) U_\nu(n + \hat{\mu}) U_\mu^\dagger(n + \hat{\nu}) U_\nu(n) \right)$$

## 2.4 Wilson twisted mass Fermionen

Aufgrund der Diskretisierung der Raumzeit treten sogenannte Diskretisierungsfehler auf. Um diese Fehler zu reduzieren nutzt man die Wilson twisted mass Gitter QCD. Dabei nutzt man folgende Wirkung für das entartete leichte Quark Duplett (u,d) und das schwere Quark Duplett (c,s):

$$S_{\text{light}}[\chi^{(l)}, \bar{\chi}^{(l)}, U] = \sum_x \bar{\chi}^{(l)}(x) (D_W + m + i\mu\gamma_5\tau_3) \chi^{(l)} \quad (3)$$

$$S_{\text{heavy}}[\chi^{(h)}, \bar{\chi}^{(h)}, U] = \sum_x \bar{\chi}^{(h)}(x) (D_W + m + i\mu_\sigma\gamma_5\tau_1 + \mu_\delta\tau_3) \chi^{(h)} \quad (4)$$

Wobei gilt:

$$D_W = \frac{1}{2}\gamma_\mu (\Delta_\mu^f + \Delta_\mu^b) + \frac{ar}{2}\Delta_\mu^f\Delta_\mu^b$$

Dabei wird die physikalische Basis  $\Psi, \bar{\Psi}$  durch die twisted mass Basis  $\chi, \bar{\chi}$  ersetzt. Der Zusammenhang zwischen den Basen ist folgender:

$$\Psi = \exp(i\gamma_5\tau_3\frac{\omega}{2})\chi, \bar{\Psi} = \exp(i\gamma_5\tau_3\frac{\omega}{2})\bar{\chi}$$

Dabei ist jedes  $\mu$  die sogenannte twisted mass und  $\tau_1$  und  $\tau_3$  die Pauli-Matrix im Farbraum. Diese Formulierung entfernt die Diskretisierungsfehler in der Ordnung  $O(a)$ . Außerdem tritt durch diese Formulierung das ‘‘Fermion-Doubling’’ nicht mehr auf. Diese Vorteile erkaufte man jedoch damit, dass die Parität keine Symmetrie mehr ist.

## 2.5 Klassifizierung von Hadronen und Erzeugungsoperatoren

Aus Kapitel 2.4 ist bekannt, dass man mit Hilfe von Korrelatoren die Massen von Hadronen bestimmen kann. Essentiell für die Berechnung von Korrelatoren sind die dabei genutzten Erzeugungsoperatoren. Dabei ist darauf zu achten, dass die Erzeuger genau die Quantenzahlen erzeugen, die man untersuchen möchte [Wag15].

### 2.5.1 Klassifizierung von Hadronen

Heutzutage werden Hadronen durch bekannte Quantenzahlen aus der Quantenfeldtheorie klassifiziert [Wag15]:

1. Isospin I;  $I_z$ .

$I_z = \frac{1}{2}$  für das up-Quark.

$I_z = -\frac{1}{2}$  für das down-Quark.

2. Gesamtdrehimpuls J.

Bosonen: J ist gerade.

Fermionen:  $J$  ist ungerade.

3. Parität  $P = \pm 1$ .

4. Ladungskonjugation  $C = \pm 1$ .

5. Weitere Flavourquantenzahlen: Strangeness, Topness, Bottomness, Charm

### 2.5.2 Ladungskonjugation C

Die Ladungskonjugation transformiert Teilchen in Antiteilchen. Die Wirkung ist definiert durch [GL]:

$$\begin{aligned}\Psi(n) &\xrightarrow{C} \Psi(n)^C = C^{-1}\bar{\Psi}(n)^T \\ \bar{\Psi}(n) &\xrightarrow{C} \bar{\Psi}(n)^C = -\Psi(n)^T C\end{aligned}$$

Außerdem gilt für  $C$ :

$$C\gamma_\mu C^{-1} = \gamma_\mu^T$$

Eine oft genutzte explizite Darstellung von  $C$  ist für die chirale Darstellung der Gamma-Matrizen gegeben durch:

$$C = i\gamma_2\gamma_0$$

### 2.5.3 Parität P

Die Parität entspricht einer Raumspiegelung mit folgender Wirkung [GL]:

$$\begin{aligned}\Psi(\vec{n}, n_0) &\xrightarrow{P} \Psi(\vec{n}, n_0)^P = \gamma_0\Psi(-\vec{n}, n_0) \\ \bar{\Psi}(\vec{n}, n_0) &\xrightarrow{P} \bar{\Psi}(\vec{n}, n_0)^P = -\bar{\Psi}(-\vec{n}, n_0)\gamma_0\end{aligned}$$

### 2.5.4 Erzeugungsoperatoren für Mesonen

Zur Klassifizierung von Mesonen nutzt man zusätzlich zu den Flavourquantenzahlen die Quantenzahlen  $J$ ,  $P$  und  $C$ . Die Notation dafür ist meist  $J^{PC}$ . Wie werden jedoch genau die gewünschten Quantenzahlen im Erzeugungsoperator verwirklicht?

Es stellt sich heraus, dass ein allgemeiner Mesonerzeuger von folgender Form ist [GL]:

$$O_M = \bar{\Psi}^{(f_1)}(n)\Gamma\Psi^{(f_2)}(n)$$

$\Gamma$  ist hierbei eine beliebige Kombination aus den bekannten Gamma-Matrizen.  $\Gamma$  ist schließlich auch dafür verantwortlich, welche Quantenzahlen durch den Erzeuger angeregt werden. In der folgenden Tabelle sind einige Möglichkeiten für  $\Gamma$  mit den dazu gehörigen Quantenzahlen und Teilchen aufgelistet.

Typ	$J^{PC}$	$\Gamma$	Teilchen
Skalar	$0^{++}$	$\gamma_0$	$f_0; a_0$
Pseudoskalar	$0^{-+}$	$\gamma_5; \gamma_0\gamma_5$	$\pi^\pm$
Vektor	$1^{--}$	$\gamma_i; \gamma_0 \gamma_i$	$\rho^\pm$
Axial-Vektor	$1^{+-}$	$\gamma_i\gamma_5$	$f_1; a_1$
Tensor	$1^{+-}$	$\gamma_i\gamma_j$	$h_1$

Tabelle 1: Beispiele  $\Gamma$  [GL]

Im folgenden werde ich anhand des Pionerzeugers erläutern, wie man die Quantenzahlen findet. Der Erzeuger für das  $\pi^\pm$  ist gegeben durch [GL]:

$$O_{\pi^+}(n) = \bar{d}(n)\gamma_5 u(n)$$

$$O_{\pi^-}(n) = \bar{u}(n)\gamma_5 d(n)$$

Damit ergibt sich für Parität und Ladungskonjugation folgendes Bild:

$$O_{\pi^+}(\vec{n}, n_0)^P = \bar{d}(-\vec{n}, n_0)\gamma_0\gamma_5\gamma_0 u(-\vec{n}, n_0) = -\bar{d}(-\vec{n}, n_0)\gamma_5 u(-\vec{n}, n_0) = -O_{\pi^+}$$

$$O_{\pi^-}(\vec{n}, n_0)^P = \bar{u}(-\vec{n}, n_0)\gamma_0\gamma_5\gamma_0 d(-\vec{n}, n_0) = -\bar{u}(-\vec{n}, n_0)\gamma_5 d(-\vec{n}, n_0) = -O_{\pi^-}$$

$$O_{\pi^+}(n)^C = -\bar{d}(n)^T C\gamma_5 C^{-1} u(n)^T = -d(n)^T \gamma_5^T \bar{u}(n)^T = O_{\pi^-}$$

$$O_{\pi^-}(n)^C = -\bar{u}(n)^T C\gamma_5 C^{-1} d(n)^T = -u(n)^T \gamma_5 \bar{d}(n)^T = O_{\pi^+}$$

Bei der Paritätstransformation muss beachtet werden, dass man sich in dem Beispiel den Impuls des Pions 0 gesetzt hat und daher über alle Raumpunkte summieren muss. Daher spielt der Übergang  $\vec{n}$  zu  $-\vec{n}$  keine Rolle für das Ergebnis. Der Isospin wird durch die Kombination der Up- und Down-Quarks realisiert und der Gesamtdrehimpuls folgt ebenfalls aus der Kombination der beiden Quarks.

### 2.5.5 Erzeugungsoperatoren für Baryonen

Die Vorgehensweise für Baryonen ist äquivalent zur Vorgehensweise bei den Mesonen. Die Unterschiede bestehen lediglich darin, dass zusätzlich zu den Flavourquantenzahlen nur J und P genutzt werden, und natürlich, dass Baryonen aus einer Kombination aus drei und nicht zwei Quarks bestehen. Eine Möglichkeit für einen Baryonerzeuger ist durch

folgenden Ausdruck gegeben [GL]:

$$O_B = \epsilon_{abc} \Gamma_1 \Psi_1^{(f_a)}(n) [\Psi_2^{(f_b)}(n) C \Gamma_2 \Psi_3^{(f_c)}(n)]$$

Auch hier werden die Flavourquantenzahlen durch die Wahl der Quarks festgelegt. Für die Quantenzahlen P und J spielen nun wieder die beiden  $\Gamma$  eine wichtige Rolle.

## 2.6 Bestimmung von Massen

### 2.6.1 Korrelatoren

Wie bereits in Teil 2.2.1 beschrieben, können Vakuumerwartungswerte mit Hilfe des Pfadintegralformalismus berechnet werden:

$$\begin{aligned} \langle \Omega | \hat{O}^\dagger(t_1) \hat{O}(t_0) | \Omega \rangle &= \frac{1}{Z} \int \mathcal{D}\Psi \mathcal{D}\bar{\Psi} \mathcal{D}A \hat{O}^\dagger(t_1) \hat{O}(t_0) \exp(-S_E[\Psi, \bar{\Psi}, A]) \\ &= \langle \Omega | e^{\hat{H}t_1} \hat{O}^\dagger(t_1) e^{(-\hat{H}t_1)} e^{\hat{H}t_0} \hat{O}(t_0) e^{(-\hat{H}t_0)} | \Omega \rangle \end{aligned}$$

Diese 2-Punkt-Funktionen nennt man auch Korrelationsfunktionen. Wenn man in Korrelationsfunktionen eine vollständige Eigenbasis zu einem Hamiltonoperator einschiebt, lassen sich aus diesen Korrelationsfunktionen Observablen wie Massen bzw. Energien von Eigenzuständen bestimmen:

$$C(t_1, t_0) = \langle \Omega | \hat{O}^\dagger(t_1) \hat{O}(t_0) | \Omega \rangle = \sum_n |\langle n | \hat{O}(0) | \Omega \rangle|^2 \exp((E_n - E_\Omega)\Delta t), \text{ mit } \Delta t = t_1 - t_0$$

Wenn man nun  $\Delta t$  sehr groß werden lässt, bricht die Summe zusammen, da die Terme, die nicht dem Grundzustand entsprechen, aufgrund des größeren Ausdrucks im Argument der Exponentialfunktion deutlich stärker unterdrückt werden. Es bleibt also schließlich nur noch der Term mit dem Grundzustand übrig:

$$\lim_{t \rightarrow \infty} \langle \Omega | \hat{O}^\dagger(t_1) \hat{O}(t_0) | \Omega \rangle = |\langle 0 | \hat{O}(0) | \Omega \rangle|^2 \exp((E_0 - E_\Omega)\Delta t)$$

Man kann nun aus dem exponentiellen Verhalten der Korrelationsfunktion bei großen Zeiten auf die Grundzustandsenergie des erzeugten Zustandes schließen. Die Energiedifferenz  $E_0 - E_\Omega$  kann mit der Masse des beobachteten Zustandes identifiziert werden. Es gilt also

$E_0 - E_\Omega = m$ . Zunächst wird die effektive Masse eingeführt, die für große  $\Delta t$  gegen die eigentliche Masse des Grundzustandes konvergiert:

$$m_{\text{eff}} = \ln\left(\frac{C(t)}{C(t+1)}\right) = E_n - E_\Omega$$

$$\lim_{t \rightarrow \infty} m_{\text{eff}} = m$$

Diese Betrachtung ist jedoch nur im Kontinuum richtig. Auf dem Gitter ergibt sich aufgrund der periodischen Randbedingungen ein anderes Bild. Da hier die Hadronen in der Zeit vorwärts wie rückwärts laufen können, wird der Korrelator nicht von einer, sondern von einer Summe aus zwei Exponentialfunktionen beschrieben. Aus dieser Summe ergibt sich schließlich ein "Cosinus Hyperbolicus".

### 2.6.2 Korrelator-Matrizen

Um die Massen von angeregten Zuständen zu bestimmen, fittet man eine Summe von Exponentialfunktionen an die numerischen Daten des Korrelators. In der Praxis stellt sich jedoch heraus, dass schon bei zwei Exponentialfunktionen die Fits sehr instabil werden und damit die Aussagekraft der Ergebnisse zweifelhaft ist.

Um dieses Problem zu umgehen, nutzt man sogenannte Korrelator-Matrizen. Dabei betrachtet man ein Set von  $N$  Operatoren  $O_j$  mit  $j \in \{1, \dots, N\}$  und berechnet alle möglichen Korrelatoren. Es ergibt sich folgender Ausdruck:

$$\begin{aligned} C_{jk}(t) &= \langle \Omega | \hat{O}_j^\dagger(t) \hat{O}_k(0) | \Omega \rangle \\ &= \sum_n \langle n | \hat{O}_j(0) | \Omega \rangle \langle n | \hat{O}_k(0) | \Omega \rangle \exp(-(E_n - E_\Omega)t) \\ &= \sum_n a_{jn}^\dagger a_{nk} \exp(-(E_n - E_\Omega)t) \end{aligned}$$

Aus dieser Matrix kann man mit verschiedenen Methoden angeregte Zustände extrahieren.

### 1.Methode: Generalisiertes Eigenwert Problem (GEP)

Das GEP ist gegeben durch:

$$C_{jk}(t)v_k^{(n)}(t, t_0) = C_{jk}(t_0)v_k^{(n)}(t, t_0)\lambda^{(n)}(t, t_0); n = 1, \dots, N; t > t_0$$

Man definiert analog zum einzelnen Korrelator die effektive Masse durch:

$$m_{\text{eff}}^{(n)}(t, t_0) = \ln\left(\frac{\lambda^{(n)}(t, t_0)}{\lambda^{(n)}(t+a, t_0)}\right)$$

Diese Methode liefert Ergebnisse, die sehr gut mit den Experimenten übereinstimmen. Beim GEP gibt es jedoch das Problem, dass es nur angewendet werden kann, wenn man alle Komponenten der Korrelatormatrix kennt. Das heißt aber auch, dass Matrixelemente in das Ergebnis einfließen, die möglicherweise grobe Fehler haben. Da sich diese Arbeit mit einer anderen Möglichkeit der Gewinnung von angeregten Zuständen beschäftigt, wird nicht näher auf das GEP eingegangen.

## 2.Methode: $\chi^2$ minimierender Fit

Bei dieser Methode, auf der diese Arbeit fußt, wählt man nun folgenden Ansatz. Man fittet an jeden einzelnen Korrelator eine Summe aus Exponentialfunktionen, wobei einzelne Parameter zwischen den Korrelatoren geteilt sind. Das heißt, dass man einen globalen Fit durchführt. Die Fitbedingung ist, das  $\chi^2$  zu minimieren, wobei  $\chi^2$  gegeben ist durch:

$$\chi^2 = \sum_{j,k=1}^N \sum_{t=t_{\min}}^{t_{\max}} \left( \frac{C_{jk}^{\text{Gitter}}(t) - C_j^{\text{fit}} k(t)}{\sigma_{jk}} \right)^2$$

Dies ist notwendig, da die Korrelatoren nicht unabhängig voneinander sind, wie man im folgenden Beispiel leicht sehen kann. Das Beispiel ist eine 2x2 Matrix und wir betrachten nur eine Exponentialfunktion. Schreibt man nun die einzelnen Komponenten der Matrix aus ergibt sich folgendes Bild.

$$C_{11}(t) = a_1^\dagger a_1 \exp(-(E_0 - E_\Omega)t) \quad (5)$$

$$C_{12}(t) = a_1^\dagger a_2 \exp(-(E_0 - E_\Omega)t) \quad (6)$$

$$C_{21}(t) = a_2^\dagger a_1 \exp(-(E_0 - E_\Omega)t) \quad (7)$$

$$C_{22}(t) = a_2^\dagger a_2 \exp(-(E_0 - E_\Omega)t) \quad (8)$$

Man erkennt, dass die Komponenten nicht unabhängig voneinander sind. So sind die Komponenten  $C_{12}$  und  $C_{21}$  äquivalent, da man wenn man  $a_j$  kennt natürlich auch  $a_j^\dagger$  kennt. Man sieht aber auch, dass die Diagonalkomponenten und die Offdiagonalkomponenten nicht unabhängig voneinander sind, da sie die Faktoren  $a_j$  und  $a_j^\dagger$  teilen.

Die Vorteile dieser Methode sind, dass man einzelne Komponenten aus dem Fit ausschließen kann. Das heißt, dass man nicht mehr die vollständige Matrix kennen muss, um Lösungen zu erhalten. Außerdem kann man sehr schlechte Korrelatoren aus dem Fit ausschließen, um so die Ergebnisse zu verbessern. Auf die Qualität und Gültigkeit der Ergebnisse wird im Kapitel 3 eingegangen.

## 2.7 $\chi^2$ minimierende Fits

Das Ziel beim fitten von Funktionen an gegebene Datenpunkte ist, eine Funktion zu finden die die gegebenen Datenpunkte bestmöglich beschreibt. Dazu muss man die funktionale Abhängigkeit der Datenpunkte zu Damit man die bestmögliche Funktion findet, stellt man eine Minimierungsbedingung auf und löst das Minimierungsproblem. Eine Möglichkeit wäre beispielsweise:

$$M = \sum_{k=1}^N (y_k - f(x_k, a_i))^2$$

Dabei sind die  $y_k$  die Datenwerte an den Stellen  $x_k$  und die  $a_i$  die zu bestimmenden Parameter. Mit dieser Methode findet man Funktionen die die Datenpunkte sehr gut annähern. Ein Problem dieser Methode ist, dass die Fehler mit denen die Datenpunkte behaftet sind nicht in den Fit eingehen. Dies ändert sich, wenn man sogenannte  $\chi^2$  minimierende Fits nutzt. Solche sind definiert durch [Ber80]:

$$\chi^2 = \sum_{k=1}^N \left( \frac{y_k - f(x_k, \dots, a_i, \dots)}{\sigma_k} \right)^2$$

In diesem Fall wird jede Differenz noch durch den Fehler des Datenpunktes  $\sigma_k$  geteilt. Eine solche Bedingung führt dazu, dass Datenpunkte mit geringem Fehler stärker in den Fit eingehen als Datenpunkte mit großem Fehler.

Um nun das Minimum zu finden und damit die passenden Parameter, sucht man die Nullstellen der Ableitung bzgl. der variablen Parameter.

Es gilt:

$$0 = \sum_{k=1}^N \left( \frac{y_k - f(x_k, \dots, a_i, \dots)}{\sigma_k} \right)^2 \left( \frac{\partial f(f(x_k, \dots, a_i, \dots))}{\partial a_i} \right)$$

Aus praktischen Gründen nutzt man zur Lösung solcher Minimierungsprobleme Algorithmen und lässt diese von Computern ausführen. Dabei können jedoch Probleme auftauchen. So hängt die Laufzeit erheblich von der Wahl der Startparameter ab, das kann sogar so weit gehen, dass der Minimierungsalgorithmus je nach Wahl der Startparameter mal ein Minimum findet, mal nicht. Außerdem ist nicht sicher gestellt, ob das jeweilige Programm ein globales oder nur lokales Minimum gefunden hat. Daher sollte man den Fit mehrmals mit unterschiedlichen Startparametern ausführen, um sicher zu gehen, dass man ein vertrauenswürdigen Ergebnis bekommen hat.

## 2.8 Binning

Da in Daten aus Monte-Carlo-Simulationen sogenannte Autokorrelationen auftreten, muss man Methoden entwickeln, um diese so gut wie möglich zu reduzieren. Eine Möglichkeit ist



das Binning. Dabei wird der gesamte Datensatz in Intervalle eingeteilt und man nutzt dann die Mittelwerte dieser Intervalle zum weiterverarbeiten der Daten. Für einen Datensatz mit den Werten  $x_i$  mit ( $i = 1, 2, 3, \dots, N$ ) und der Binngröße  $n$  folgt:

$$x'_j = \frac{1}{n} (x_{1+j \cdot n} + x_{2+j \cdot n} + \dots + x_{n+j \cdot n}) \text{ mit } j = 0, \dots, \frac{N}{n}$$

Hierbei muss darauf geachtet werden, dass  $\frac{N}{n}$  ohne Rest aufgeht. Ansonsten ist die obige Formel nicht richtig. Dieses Vorgehen sorgt dafür, dass die Neue nun geringere Anzahl von Daten untereinander deutlich weniger korreliert sind. Ein weiterer Vorteil ist, dass durch dieses Vorgehen die Datenmenge deutlich reduziert werden kann.

## 2.9 Die delete-1-Jackknife-Methode

Um die Qualität eines Mittelwertes  $\bar{X}$  der sich aus den Einzelwerten ( $X_1, \dots, X_N$ ) ergibt zu bestimmen, kann man auf die sogenannte Jackknife Methode zurückgreifen. Die Jackknife Methode basiert darauf, aus der Gesamtprobe  $n$  zusätzliche Stichproben zu generieren. Dies wird dadurch erreicht, dass man aus der Gesamtprobe  $n-1$  Werte heraus nimmt und über diese den Mittelwert berechnet. Dabei lässt man jeden Einzelwert einmal aus der Probe herausfallen und damit ergeben sich  $n$  Stichproben. Die  $i$ -te Stichprobe ist von der Form ( $X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_N$ ) mit Mittelwert  $\bar{X}_i$ .

Für die Mittelwerte und die Standardabweichung folgt damit [Kna07]:

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i$$

$$\bar{X}_i = \frac{1}{N-1} \sum_{k=1, k \neq i}^{N-1} X_k$$

$$\sigma(X) = \sqrt{\frac{N-1}{N} \sum_{i=1}^N (\bar{X} - \bar{X}_i)^2}$$

## 3 Programmierarbeit

Zur Bestimmung der Hadronmassen nutze ich in dieser Arbeit die in Kapitel 2.6.2 vorgestellte Methode der  $\chi^2$  minimierenden Fits. Für diese Aufgabe war es notwendig, ein eigenes, voll funktionsfähiges Programm zu schreiben. Das Programm ist komplett in der Programmiersprache C++ verfasst und basiert zu großen Teilen auf externen Bibliotheken wie der Root Library des CERN.

### 3.1 Anforderungen des Programmes

Im folgenden werde ich die Hauptanforderungen an das Programm auflisten, um eine erste Übersicht über die Funktionen des Programmes zu liefern.

1. Ausführen simultaner Fits an alle Elemente der Korrelator-Matrix, mit besonderem Augenmerk darauf, dass die einzelnen Korrelatoren nicht unabhängig sind. Das heißt, die Fitparameter dürfen nicht unabhängig voneinander für jeden Korrelator ermittelt werden.
2. Die Möglichkeit, einzelne Elemente aus der Korrelator-Matrix aus den simultanen Fits auszuschließen, wie in Kapitel 2.6.2 erläutert.
3. Die Möglichkeit, die Fitbereiche für jedes Matrixelement unabhängig von den anderen einzustellen.
4. Die Möglichkeit, unterschiedliche Anzahlen von Exponentialfunktionen an die Daten zu fitten.
5. Routine zum Ausführen der Jackknife-Methode aus Kapitel 2.9, um seriöse Aussagen über die Fehler der Ergebnisse treffen zu können.

### 3.2 Ablauf des Programms und Flussdiagramm

Zunächst wird das Programm per Kommandozeile gestartet. Zusätzlich muss der Dateiname des auszuwertenden Optionfiles angegeben werden. Wenn das geschehen ist, läuft das Programm ohne weitere Bedienung durch den User. Zunächst werden die Daten eingelesen und in ein für das Programm nutzbares Format gebracht. Das heißt, es wird das Binning durchgeführt und im gleichen Schritt werden die für die Jackknife Methode benötigten Subsamples erzeugt und gespeichert.

Für jedes Subsample und die gesamte Probe wird nun die Fitfunktion aufgerufen, die Funktionen werden je nach Wunsch (spezifiziert im Optionsfile) erzeugt und das Sharing zwischen den Parametern wird sichergestellt. Die Ergebnisse für die Energien werden gespeichert und an die Jackknife-Funktion weitergeleitet.

Diese berechnet dann den Fehler und gibt den Mittelwert mit Fehler als Ergebnis aus. In der Fitfunktion wird gleichzeitig für die gesamte Probe die komplette Matrix mit den Datenpunkten und den gefitteten Funktionen in ein PDF-File geplottet.

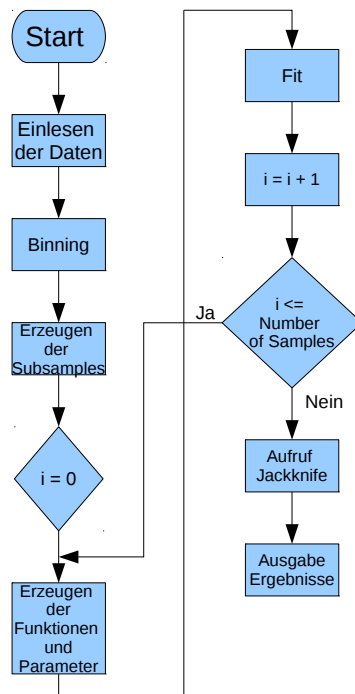


Abbildung 1: Flussdiagramm

### 3.3 Simultanes Fitten

Zur Durchführung der simultanen Fits benutze ich die Root Library des CERN. Im folgenden Abschnitt werde ich erläutern, wie die Funktionen und das Teilen der Parameter erzeugt werden und dadurch das simultane Fitten sichergestellt wird. Dafür muss man sich zunächst klar machen, wie viele Parameter es insgesamt gibt und welcher Parameter

in welchem Matrix-Element vorkommt. Ein allgemeines Matrixelement einer Korrelator-Matrix ist bereits bekannt:

$$C_{jk}(t) = \sum_n a_{jn}^\dagger a_{nk} \exp(-(E_n)t)$$

Was sich umschreiben lässt zu:

$$C_{jk}(t) = \sum_n [\operatorname{Re}(a_{jn}) - i \operatorname{Im}(a_{jn})][\operatorname{Re}(a_{nk}) + i \operatorname{Im}(a_{nk})] \exp(-(E_n)t)$$

Man sieht also, dass in jedem Matrixelement pro Exponentialfunktion 5 verschiedene Parameter vorkommen. Die Energie und die Real- bzw. Imaginärteile der beider Faktoren der Amplitude.

Für das Programm muss außerdem beachtet werden, dass die Korrelatoren selber daher einen Real- und Imaginärteil haben. Dies spielt gerade in den Funktionen `GenerateFunction` und `GenerateParameters` die nachfolgend vorgestellt werden eine Rolle.

Für die Gesamtzahl der Parameter ergibt sich aus der Tatsache, dass die Matrixkomponenten nicht unabhängig voneinander sind, die Formel:

$$N_{par} = \text{NumberOfFunctions} \times (2 \times \text{DimensionOfMatrix} + 1)$$

Für das Erzeugen der Funktionen in der `Fit`-Routine müssen diese als Strings an diese weitergegeben werden. Hier sieht man wie diese Strings konstruiert werden.

```

1 template<typename FloatingPoint>
2 string CorrelatorMatrix<FloatingPoint>::GenerateFunction(int NumberOfFunctions,
3     bool ImaginaryPart = false) const { //Erzeugen der Funktionenstrings
4     std::string FunctionString;
5     int i=0;
6     for(int m = 1; m <= NumberOfFunctions; ++m) {
7         FunctionString.append("(" + std::to_string(i++) + ")*
8         [" + std::to_string(i++) + "]" + ((ImaginaryPart)? "-":"+") +
9         [" + std::to_string(i++) + "]*[" + std::to_string(i++) + "]"
10        *exp(-[" + std::to_string(i++) + "]*x)");
11        if( m < NumberOfFunctions) FunctionString.append("+");
12    }
13    return FunctionString;
14 }
```

Ein solcher String hätte für den Realteil eines Korrelators z.B. folgende Form:

$$([0]*[1] + [2]*[3]) * \exp([4]*x)$$

Man sieht also es gibt 5 Parameter. Für jede weitere Exponentialfunktion wäre die Form die gleiche, nur die Zahlen würden sich verändern.

Das bestimmende Element für die globalen simultanen Fits ist jedoch die Definition der Parameter. Diese werden in einer Liste gespeichert, die an die in der Root definierten globalen Chi2 Funktion weitergegeben werden. Das heißt, dass die Information welche Parameter in welchem Korrelator vorkommen in dieser Liste hinterlegt ist. In der nächsten Abbildung wird gezeigt wie diese Parameter-Liste erzeugt wird.

```

1 template<typename FloatingPoint>
2 std::vector<int> CorrelatorMatrix<FloatingPoint >::GenerateParameters(int Dimension,
3                               int NumberOfFunctions, int Row, int Column,
4                               bool ImaginaryPart = false)const {
5     Row +=1;
6     Column +=1;
7     std::vector<int> Parameters;
8     for(int m = 0; m < NumberOfFunctions; ++m) {
9         if (! ImaginaryPart) {
10            Parameters.push_back((2 * Dimension + 1) + (2 * Dimension + 1) * m - 1);
11            Parameters.push_back((Column + Dimension) + (2 * Dimension + 1) * m - 1);
12            Parameters.push_back((Row + Dimension) + (2 * Dimension + 1) * m - 1);
13            Parameters.push_back(Column + (2 * Dimension + 1) * m - 1);
14            Parameters.push_back(Row + (2 * Dimension + 1) * m - 1);
15        } else {
16            Parameters.push_back((2 * Dimension + 1) + (2 * Dimension + 1) * m - 1);
17            Parameters.push_back(Column + (2 * Dimension + 1) * m - 1);
18            Parameters.push_back((Row + Dimension) + (2 * Dimension + 1) * m - 1);
19            Parameters.push_back((Column + Dimension) + (2 * Dimension + 1) * m - 1);
20            Parameters.push_back(Row + (2 * Dimension + 1) * m - 1);
21        }
22    }
23
24    return Parameters;
25 }

```

Im Grunde passiert hier nichts anderes, als zu schauen welche Parameter in welcher Zeile und in welcher Spalte der Matrix vorkommen. Allen gleichen Parametern wird dann die gleiche Zahl zugeordnet.

Dadurch ist festgelegt, dass die Parameter geteilt sind. So wird für den String  $([0]*[1] + [2]*[3]) * \exp([4]*x)$  über die Liste  $[0,1,0,2,3]$  festgelegt, dass Parameter  $[0]$  und  $[2]$  der

gleiche Parameter ist.

Die Information welcher Parameter angesprochen wird, ist also die Stelle in der Liste, und der Wert an dieser Stelle sagt was für ein Parameter es ist. Das Erzeugen dieser Parameter Liste ist tatsächlich das Herzstück des Programmes obwohl es recht unscheinbar aussieht.

### 3.4 Das Optionsfile

Ein weiteres wichtiges Feature ist , die Fitdetails dem Programm übergeben zu können ohne jedes mal neu kompilieren zu müssen. Daher war es notwendig ein Optionsfile zu erstellen, das alle Anforderungen erfüllt. Diese Anforderungen sind beispielsweise die Angabe der Binningsize, die Fitbereiche und die Angabe welche Matrixelemente in den Fit einfließen sollen. Das Optionsfile und dessen Interpretation durch das Programm ist mit Hilfe des Datenformats Json implementiert. Die Form eines solchen Optionsfiles ist folgende:

```
1 {
2     "Global" : {
3         "Dateiname" : "Probe Norm Gut.dat",
4         "NumberOfFunctions" : 2,
5         "TMin" : 11,
6         "TMax" : 25,
7         "BinningSize" : 1,
8         "FitReal" : false,
9         "FitImag" : false,
10        "Energy" : [0.799, 0.98]
11    },
12    "MatrixElements" : [
13        {
14            "NumberOfMatrixelement" : 0,
15            "TMin" : 11,
16            "TMax" : 25,
17            "FitReal" : true,
18            "FitImag" : true
19        },
20        {
21            "NumberOfMatrixelement" : 1,
22            "TMin" : 11,
23            "TMax" : 25,
24            "FitReal" : true,
25            "FitImag" : true
26        },
27        {
28            "NumberOfMatrixelement" : 8,
```

```
29         "TMin" : 11,
30         "TMax" : 25,
31         "FitReal" : true,
32         "FitImag" : true
33     },
34     {
35         "NumberOfMatrixelement" : 9,
36         "TMin" : 11,
37         "TMax" : 25,
38         "FitReal" : true,
39         "FitImag" : true
40     }
41 ]
42 }
```

Der Grundaufbau des Optionsfiles ist recht simpel. Am Anfang des Files steht eine Liste mit dem Schlüsselwort "Global". In dieser Liste gibt es wiederum Schlüsselwörter denen unterschiedliche Werte zugeordnet sind. Hier sieht man schon an der Namensgebung was man alles angeben kann. Die zweite Liste ist dann eine Liste mit der man jedes Matrixelement einzeln ansprechen und spezielle Vorgaben für dieses einstellen kann. Standardmäßig sind die globalen Optionen für jedes einzelne Matrixelement gesetzt. Bisher ist noch nicht implementiert einzelne Operatoren, ganze Zeilen oder Spalten ansprechen zu können.

### 3.5 Ausgabe PDF des Programmes

Zum Schluss dieses Kapitels folgt noch ein kurzes Beispiel, wie die erzeugte PDF-Datei aussehen kann. Hier kann man erkennen, dass nicht alle Elemente der Matrix in den Fit eingeflossen sind, und daher auch keine Fits in den entsprechenden Komponenten zu sehen sind. Die Realteile der Korrelatoren sind rot, die Imaginärteile blau und die Fehlerbalken schwarz. Die durchgezogenen Linien sind die gefitteten Funktionen während die Punkte gerade den numerischen Datenpunkten entsprechen.

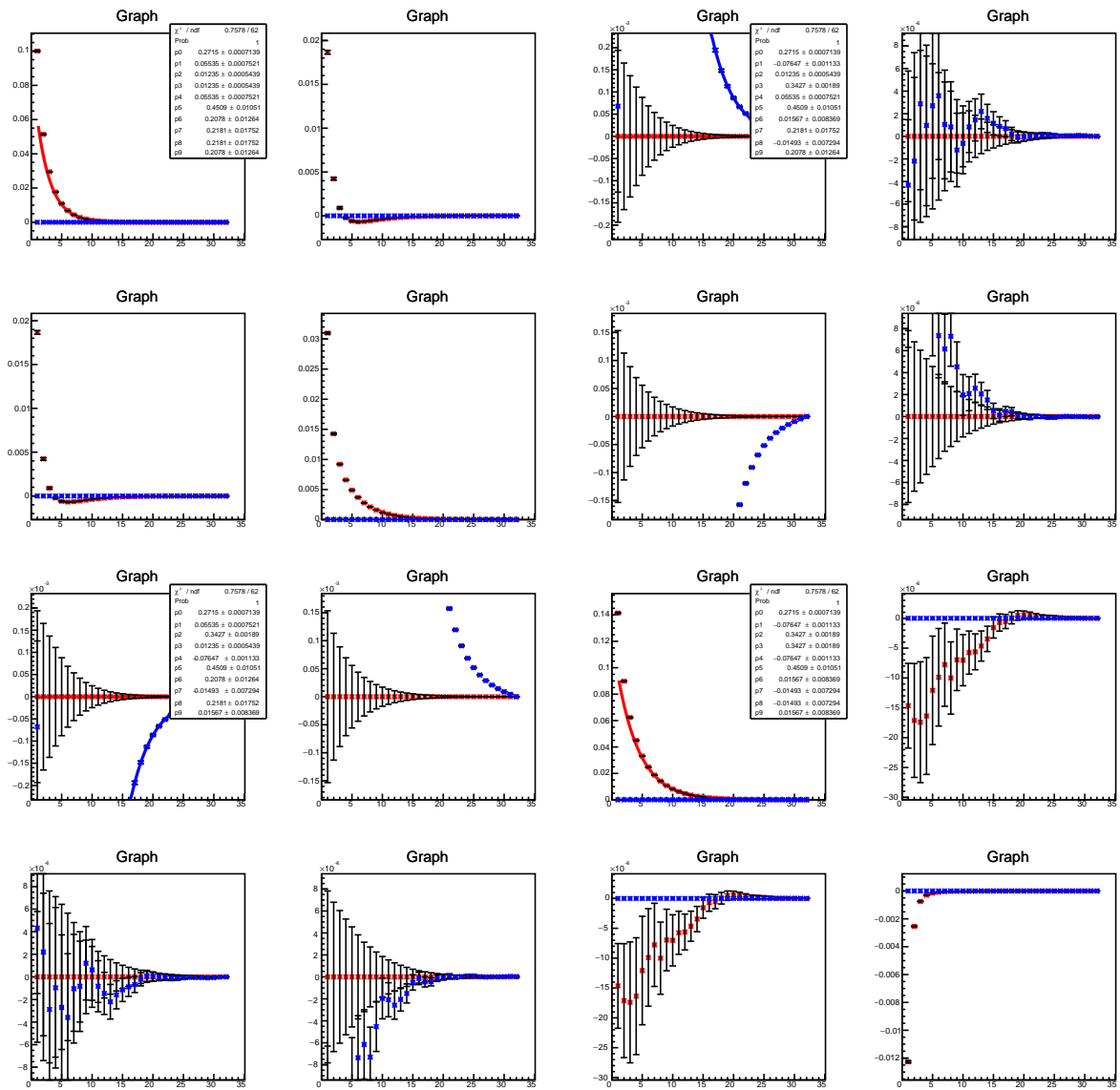


Abbildung 2: Ausgabe des Programmes



### 3.6 Probleme des Programms und mögliche Lösungen

Bereits in Kapitel 2.7 wurde kurz darauf eingegangen, dass beim Ausführen von Fitalgorithmen fehlerhafte Ergebnisse möglich sind.

In meinem speziellen Programm konnte ich einige dieser Probleme lösen andere wiederum nicht.

Das größte Problem ist, dass der Fitter nicht unbedingt das korrekte globale Minimum findet, falls die Startparameter zu schlecht gewählt sind. Die Lösung für dieses Problem ist, schlicht die Startparameter so zu wählen, dass sie sehr nahe am gesuchten Minimum liegen. Das heißt man führt einen "educated guess" durch und initialisiert die Startparameter. Trotzdem ist das keine Garantie dafür, dass man das globale Minimum findet wie in folgendem Bild deutlich wird.

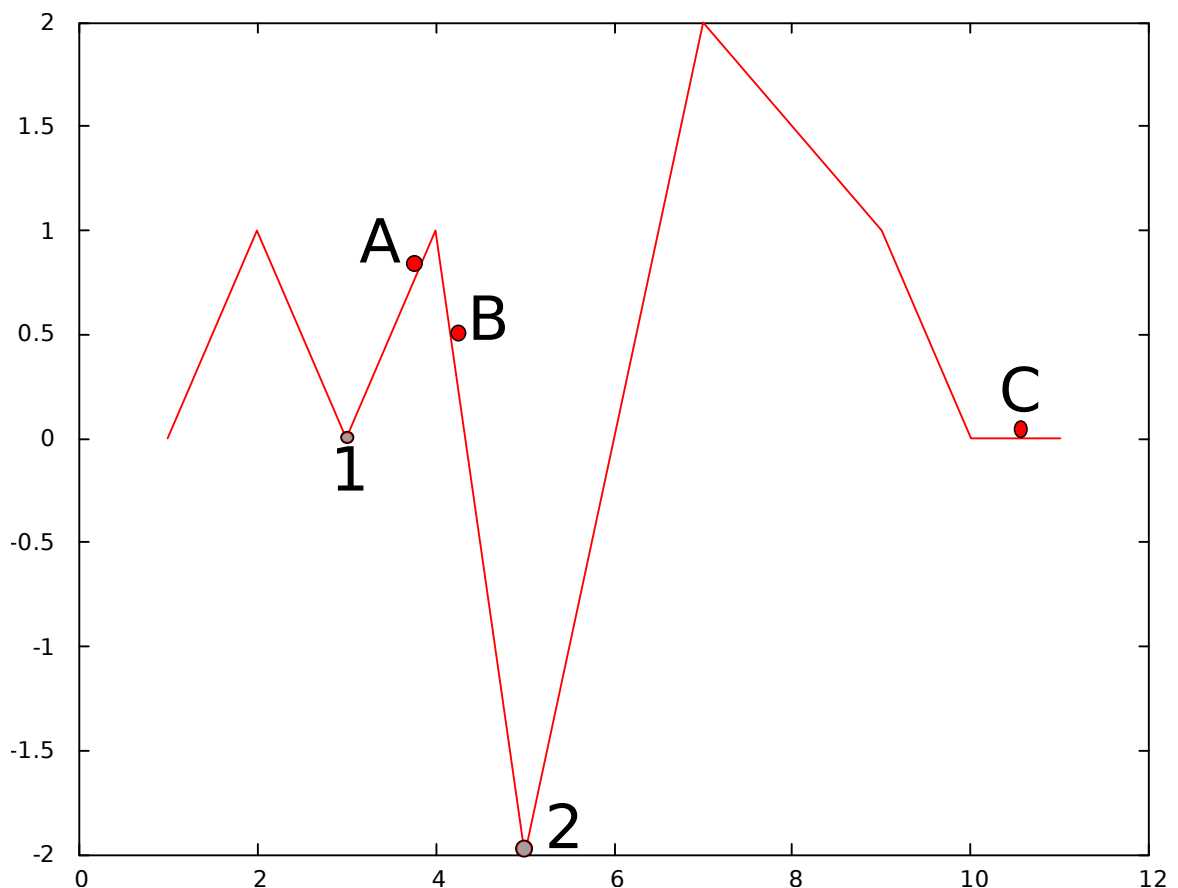


Abbildung 3: Minimierungsprobleme Beispiel

In dem Bild ist eine vereinfachte Version einer  $\chi^2$ -Funktion zu sehen. Diese soll nun minimiert werden. Man sieht zwei lokale Minima ( mit 1 bzw. 2 gekennzeichnet). Je nach

Startwerten der Parameter hat die  $\chi^2$ -Funktion unterschiedliche Startwerte (dargestellt durch die roten Kreise gekennzeichnet mit Buchstaben). Je nach verwendeten Startparametern wird nun ein anderes Minimum oder auch gar keins gefunden.

**1.Möglichkeit:** Wir starten bei Punkt A. Der Algorithmus wird das Minimum 1 finden. Dieses entspricht jedoch nicht dem tatsächlichen globalen Minimum, obwohl wir nicht weit von diesem entfernt sind.

**2.Möglichkeit:** Wir starten bei Punkt B. Der Algorithmus wird das globale Minimum finden. Dies ist ein vernünftiger Startpunkt.

**3.Möglichkeit:** Wir starten bei Punkt C. Der Algorithmus wird keinen Fortschritt machen, da der Startpunkt auf einer Ebene liegt. Er wird kein Minimum finden.

Man sieht also es gibt bereits bei einem sehr einfachen "Gebirge" mehrere Möglichkeiten ein Tal zu finden. Bei den mir zur Verfügung stehenden Daten handelt es sich meist um ungleich komplexere Probleme. Es ist also leicht vorstellbar, dass es sehr schwer werden kann das passende Minimum zu finden, wenn die Anzahl der Parameter steigt, da man dann ein mehrdimensionales Gebirge durchlaufen muss. Dies führt direkt zum nächsten Problem meines Programms.

Bei steigender Parameterzahl kann der Fit so instabil werden, dass die Aussagekraft der Ergebnisse sehr zweifelhaft teilweise sogar offensichtlich falsch sind. Dieses Problem kann man dadurch lösen, dass man einen Wertebereich für die Parameter definiert, in dem sich diese befinden müssen. Jedoch kann der Fitter auch mit dieser Hilfestellung bei manchen Problemen versagen.

Das letzte Problem auf das ich eingehen werde ist, dass der Fit an Stabilität verliert, wenn man die Anzahl der Exponentialfunktionen verändert. In der folgenden Tabelle soll dies verdeutlicht werden ohne auf die Ergebnisse an sich einzugehen, da dies im nächsten Kapitel geschieht:

Anzahl der Funktionen	2	3	4
E1	0.866792±0.0527278	0.7219±0.290456	0.662306±0.183933
E2	1.03683±0.0703945	0.731325±0.370923	0.781472±0.194824
E3		1.03533±0.294243	0.782183±0.203409
E4			1.037±0.0274122

Man sieht, dass nur der Energiewert 1.036 relativ stabil auftritt. Die anderen Werte sind so instabil, dass ihnen kaum zu vertrauen ist. Ab 5 Funktionen konvergiert der Fitter gar nicht mehr.

Dieses Problem kann man dadurch lösen, dass man die Fitreichweite für jedes Matrixelement individuell anpasst und dass man mehr Matrixkomponenten in den Fit einbezieht. Da dies die Datenlage deutlich verbessert, wird der Fitter wieder stabiler. Aber auch hier gilt, dass die Ergebnisse keine Garantie haben und noch auf Richtigkeit überprüft werden müssen.

Die oben genannten Probleme führen dazu, dass man im Umgang mit meinem Programm sehr aufmerksam sein muss und recht viel Vorwissen nutzen muss um wirklich aussagekräftige Ergebnisse zu gewinnen.

## 4 Ergebnisse und Interpretation

In dem nun folgenden Kapitel werde ich kurz einige meiner Ergebnisse vorstellen. Dabei geht es in diesem Kapitel vorrangig darum, zu überprüfen, ob der Multi-Exponential-Fit die selben Resultate liefert wie das GEP, und wird sich daher kaum mit den physikalischen Aussagen der Ergebnisse beschäftigen.

Da sich die nun folgenden Ergebnisse zu großen Teilen auf [KW15] beziehen, verweise ich für eine genauere physikalische Betrachtung auf diese Arbeit. Es werden Konfigurationen genutzt, die mit 2+1+1 dynamischen Quark Flavours generiert wurden, bereitgestellt durch die European Twisted Mass Collaboration. In der folgenden Tabelle wird das genutzte Ensemble vorgestellt.

Ensemble	$\beta$	$(L/a)^3 \times T/a$	$\mu$	$\mu_\sigma$	$\mu_\delta$	a(fm)	$m_\pi$ (MeV)	# of configurations
A30.32	1,90	$32^3 \times 64$	0.0030	0,150	0,190	0,0885	276	1200

Tabelle 2: Siehe [KW15, Kap. 4.1 Table 1]

Die verwendete  $8 \times 8$  Matrix besteht aus Korrelatoren von Mesonen, genauer gesagt werden D Mesonen bzw Charmonium untersucht. Die grobe Struktur der Operatoren ist aus 2.5.4 bekannt. Die verwendeten  $\Gamma$ s sind in 4 aufgelistet. Für eine detailliertere Übersicht verweise ich auf [KW15].

Da alle Ergebnisse in Gittereinheiten vorliegen müsste man diese noch umrechnen um sie gegebenenfalls mit experimentellen Daten abzugleichen. Auch hier verweise ich wieder auf

[KW15] für eine genauere physikalische Betrachtung.

Der Einfachheit halber werde ich nur ausgewählte Elemente der betrachteten  $8 \times 8$  Matrix zeigen um zu verdeutlichen, dass die gefitteten Funktionen die Daten gut annähern.

Für 4.1,4.2,4.3 und 4.5 ist darauf zu achten, dass bei den verwendeten Daten nicht die tatsächlichen experimentellen Werte gewonnen werden. Sondern die gewonnenen Werte in [KW15] dazu genutzt werden tatsächliche physikalische Ergebnisse zu extrapolieren. Das liegt an Effekten der Gitter QCD und der verwendeten Masse des Charm-Quarks, die größer ist als die tatsächliche Masse des Charm-Quark.

index	continuum			twisted mass lattice QCD			
	$\Gamma(n), pb$	J	PC	tb, ( $\pm, \mp$ )	tb, ( $\pm, \pm$ )	$O^S \otimes O^L \rightarrow O^J$	
1	$\gamma_5$	0	-+	pb	$\pm i \gamma_5 \times$	$A_1 \otimes A_1$	$A_1$
2	$\gamma_0 \gamma_5$		-+	$\pm i \gamma_5 \times$	pb		
3	1		++	pb	$\pm i \gamma_5 \times$		
4	$\gamma_0$		+-	$\pm i \gamma_5 \times$	pb		
5	$\gamma_5 \gamma_j n_j$		--	$\pm i \gamma_5 \times$	pb	$T_1 \otimes T_1$	
6	$\gamma_0 \gamma_5 \gamma_j n_j$		-+	pb	$\pm i \gamma_5 \times$		
7	$\gamma_j n_j$		++	$\pm i \gamma_5 \times$	pb		
8	$\gamma_0 \gamma_j n_j$		++	pb	$\pm i \gamma_5 \times$		

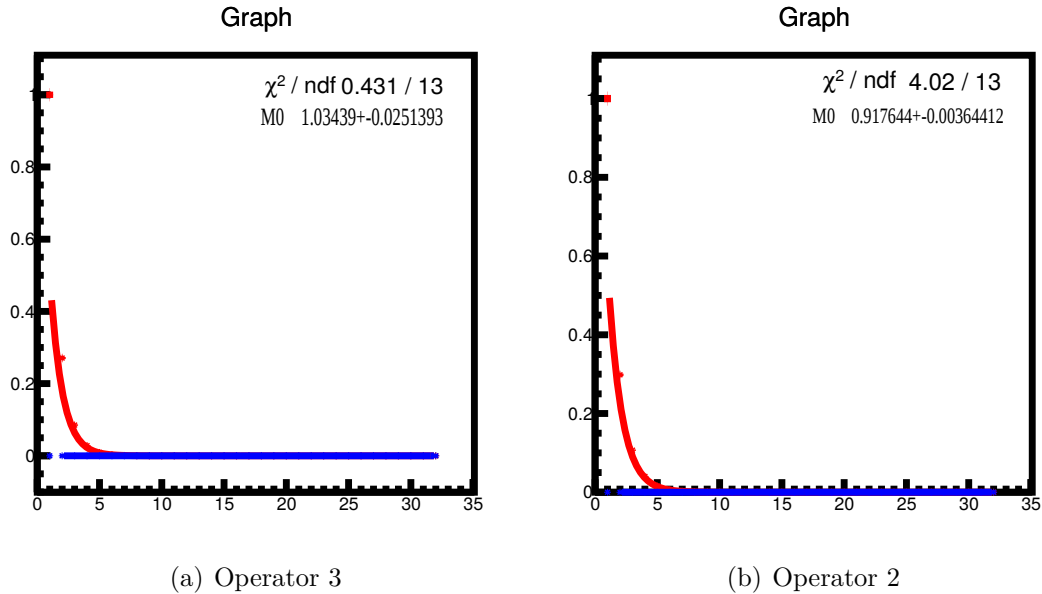
Abbildung 4: Tabelle der  $\Gamma$ s [KW15]

#### 4.1 D-Meson $1 \times 1$ Matrix : J = 0

P = - : D

P = + :  $D_0^*$

Zunächst werde ich die Grundzustandsenergie des D-Meson bestimmen. Dies tue ich indem ich aus der  $8 \times 8$  Matrix einen einzelnen Korrelator heraus nehme. Einmal nehme ich den Korrelator mit  $\Gamma = \mathbf{1}$  (linkes Bild) und einmal mit  $\Gamma = \gamma_0 \gamma_5$  (rechtes Bild). Der Zeitbereich, in dem gefittet wurde, ist von Zeitschritt 11 bis Zeitschritt 25.

Abbildung 5:  $1 \times 1$  Matrizen

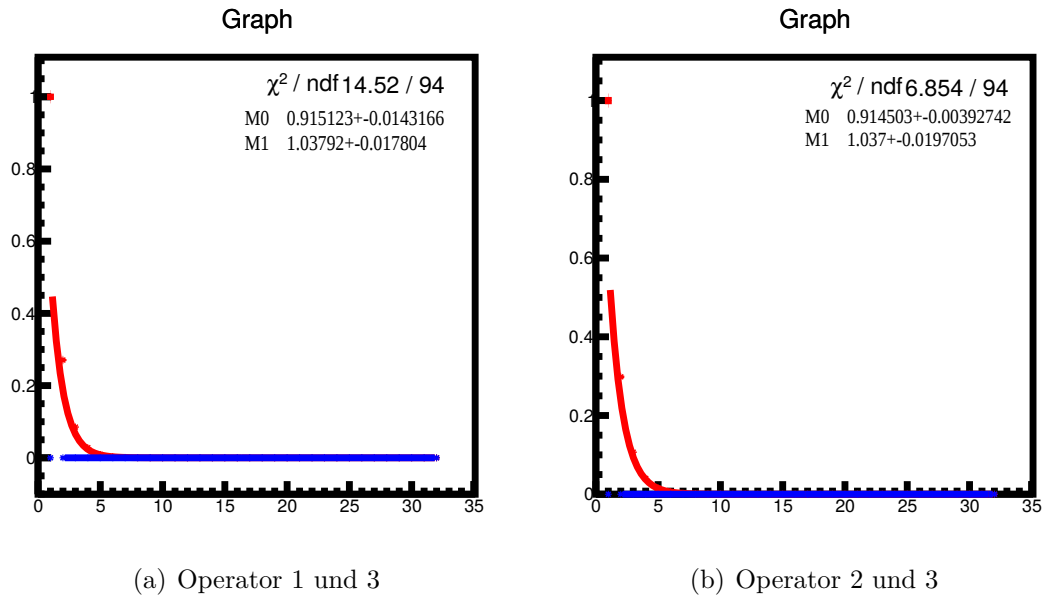
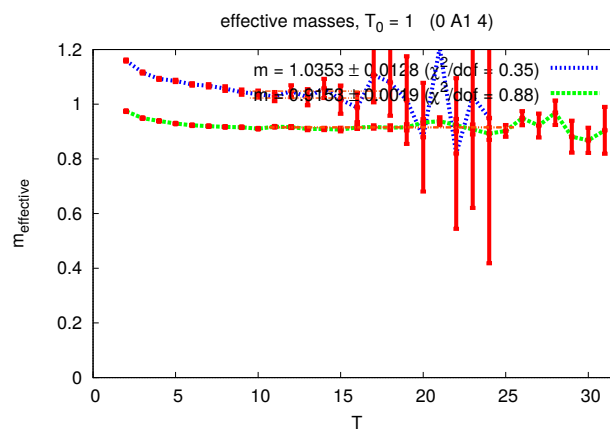
Man erkennt, dass sich beide Ergebnisse voneinander unterscheiden. Sie sind jedoch beide richtig. Es stellt sich nämlich heraus, dass es sich nicht beidesmal um den niedrigsten Zustand handelt. Das kann man damit erklären, dass der niedrigste Zustand im A1 Sektor (D-Meson) Parität - hat und der nächst höhere Zustand  $D_o^*$  Parität + hat. Aufgrund des verwendeten Twisted Mass Formalismus, liegt zwar keine strenge Trennung der Paritäts-sektoren vor 3 sie scheint hier dennoch eine Rolle zu spielen.

## 4.2 D-Meson $2 \times 2$ Matrix : $J = 0$

P = - : D

P = + :  $D_o^*$

Die folgenden Bilder sind exemplarisch, für eine  $2 \times 2$  Untermatrix. Zusätzlich wird noch das Ergebnis durch das GEP hinzugefügt. Die Matrizen sind dadurch gebildet, dass man nur die Operatoren 1 und 3 aus 4(linkes Bild) bzw. nur die Operatoren 2 und 3 aus 4(rechtes Bild) benutzt hat. Sie wurde dazu verwendet, zwei Exponentialfunktionen an die Korrelatoren zu fitten, um so die Energien der ersten beiden Zustände zu gewinnen. Der Zeitbereich, in dem gefittet wurde, ist Zeitschritt 10 bis Zeitschritt 25. Die Ergebnisse sind in 4.2 und 7 aufgezeichnet. Exemplarisch zeige ich nur jeweils den Fit in einer Matrixkomponente.

Abbildung 6:  $1 \times 1$  MatrizenAbbildung 7:  $2 \times 2$  Matrix 2 Energien GEP

Man sieht sofort, dass die Ergebnisse im Rahmen der Fehler übereinstimmen. Damit ist gezeigt, dass der Multi-Exponential-Fit für dieses Problem die gleichen Ergebnisse liefert wie das GEP.

### 4.3 D-Meson $4 \times 4$ Matrix: $J = 0$

P = - : D

P = + :  $D_o^*$

Zusätzlich habe ich noch  $4 \times 4$  Matrizen betrachtet, die jeweils aus den Operatoren 2,4,6 und 8 (linkes Bild) bzw. aus den Operatoren 1,3,5 und 7(rechtes Bild) bestanden. Der

Zeitbereich, in dem gefittet wurde, ist auch hier wieder Zeitschritt 10 bis Zeitschritt 25. Exemplarisch zeige ich nur jeweils den Fit in einer Matrixkomponente.

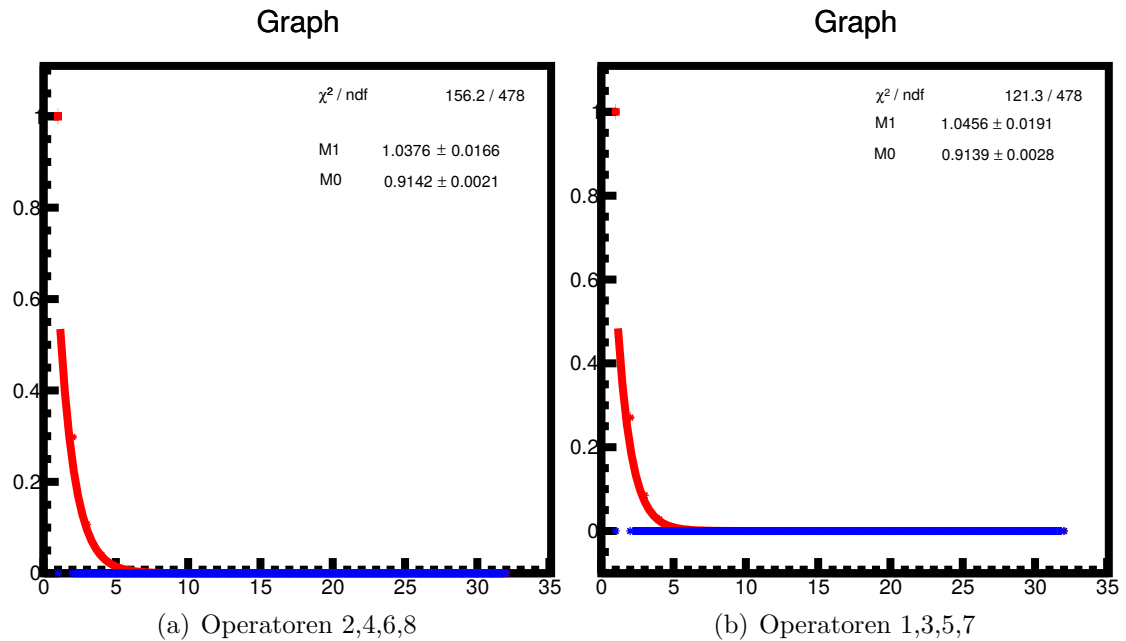


Abbildung 8:  $4 \times 4$  Matrizen

In der folgenden Tabelle sind die Ergebnisse für die ersten beiden Zustände zusammengefasst.

Genutzte Operatoren	$M_0 \cdot a(D)$	$M_1 \cdot a(D_o^*)$
<b>1</b>		$1.0344 \pm 0.0251$
$\gamma_0 \gamma_5$	$0.9176 \pm 0.0036$	
$\gamma_5$	$0.9145 \pm 0.0311$	
<b>1, <math>\gamma_5</math></b>	$0.9151 \pm 0.0143$	$1.0379 \pm 0.0178$
<b>1, <math>\gamma_0 \gamma_5</math></b>	$0.9145 \pm 0.0039$	$1.037 \pm 0.0197$
$\gamma_0 \gamma_5, \gamma_0, \gamma_0 \gamma_5 \gamma_j n_j, \gamma_0 \gamma_j n_j$	$0.9142 \pm 0.0028$	$1.0376 \pm 0.0166$
<b>1, <math>\gamma_5, \gamma_5 \gamma_j n_j, \gamma_j n_j</math></b>	$0.9139 \pm 0.0028$	$1.0456 \pm 0.0191$
<b>GEP: 1, <math>\gamma_5</math></b>	$0.9153 \pm 0.0019$	$1.035 \pm 0.0128$

Zusammengefasst, kann man also sagen, dass der Multi-Exponential-Fit(MEF) die selben Resultate liefert wie das GEP. Außerdem kann man feststellen, dass die Fehler bei steigender Matrixgröße tendenziell leicht abnehmen. Hier könnte man systematisch untersuchen ob es einen Zusammenhang gibt. Es muss jedoch nochmal darauf hingewiesen werden, dass dabei alle Lösungsvorschläge aus 3.6 genutzt wurden.

#### 4.4 D-Meson $4 \times 4$ Matrix $J = 0, P = -$

$P = - : D$

$P = + : D^*_0$

Zum Abschluss des D-Mesons stelle ich noch kurz die  $4 \times 4$  Matrix vor, die man erhält, wenn man alle Operatoren aus 4 nimmt die Parität - haben. Besonderes Augenmerk liegt dabei darauf verschiedene Ergebnisse vorzustellen, die durch verschiedene Fitbereiche und Anzahl von gefitteten Exponentialfunktionen entstanden sind. Es geht also darum zu zeigen, wie sich die Ergebnisse durch das Anpassen der Fitoptionen verändern. Daher werden hier keine Grafiken sondern nur eine Tabelle mit den Ergebnissen vorgestellt auf die dann kurz Bezug genommen wird.

Zunächst folgt die Tabelle mit nur einer Exponentialfunktion und unterschiedlichen Fitbereichen.

Zeitbereich des Fits/a	$M_0 \cdot a (D)$
11 – 25	$0.9145 \pm 0.0024$
10 – 25	$0.9147 \pm 0.0022$
9 – 25	$0.9153 \pm 0.0108$
8 – 25	$0.9159 \pm 0.0017$
7 – 25	$0.9172 \pm 0.0015$
6 – 25	$0.9192 \pm 0.0014$
5 – 25	$0.9222 \pm 0.0013$
4 – 25	$0.9266 \pm 0.0011$

Man kann erkennen, dass der Grundzustand in allen Fitbereichen sehr dominant ist. Die Ergebnisse stimmen mit denen aus den vorhergehenden Kapiteln überein. Auch für die größeren Fitbereiche mit kleinen Zeiten scheinen angeregte Zustände keine große Rolle zu spielen, da sich die Ergebnisse nur leicht ändern.

Nun folgt die Tabelle mit den gleichen Zeitbereichen aber zwei Exponentialfunktionen.



Zeitbereich des Fits/a	$M_0 \cdot a$ (D)	$M_1 \cdot a$
11 - 25	$0.9136 \pm 0.0038$	$0.9159 \pm 0.0045$
10 - 25	$0.9146 \pm 0.0039$	$0.9149 \pm 0.0040$
9 - 25	$0.9121 \pm 0.0044$	$0.9190 \pm 0.0069$
8 - 25	$0.9143 \pm 0.0019$	$1.3820 \pm 0.1076$
7 - 25	$0.9142 \pm 0.0098$	$1.4029 \pm 0.1522$
6 - 25	$0.9138 \pm 0.0021$	$1.3574 \pm 0.0982$
5 - 25	$0.9135 \pm 0.0020$	$1.3284 \pm 0.0574$
4 - 25	$0.9134 \pm 0.0018$	$1.3153 \pm 0.0267$

Hier fällt nun auf, dass ab Zeitbereich 8 - 25 ein zweites Energieniveau gefunden wird. In den vorherigen Zeitbereichen wird, zweimal fast die selbe Energie gefunden, was auf die Massendifferenz zwischen  $D^0$  und  $D^\pm$  hindeuten könnte. Weiterhin ist auffällig, dass das zweite Energieniveau nicht dem zweiten Energieniveau aus den vorhergehenden Kapiteln entspricht. Da nur Operatoren mit Parität - verwendet wurden zeigt sich der Zustand mit Parität + nicht oder nur sehr schwach, obwohl es keine strikte Trennung der Paritätssektoren gibt (Vergleich mit 4.1).

Als letztes folgt die Tabelle mit 3 Exponentialfunktionen.

Zeitbereich des Fits/a	$M_0 \cdot a$ (D)	$M_1 \cdot a$	$M_2 \cdot a$
11 - 25	Fitter konvergiert nicht		
10 - 25	Fitter konvergiert nicht		
9 - 25	Fitter konvergiert nicht		
8 - 25	$0.9144 \pm 0.0023$	$1.0746 \pm 0.1124$	$1.4521 \pm 0.1455$
7 - 25	$0.9143 \pm 0.0022$	$1.0795 \pm 0.1121$	$1.4313 \pm 0.1097$
6 - 25	$0.9138 \pm 0.023$	$1.0926 \pm 0.0273$	$1.3740 \pm 0.0799$
5 - 25	$0.9133 \pm 0.0084$	$1.1114 \pm 0.1100$	$1.3376 \pm 0.0483$
4 - 25	$0.9129 \pm 0.0022$	$1.1357 \pm 0.0118$	$1.3176 \pm 0.0294$

Hier sieht man, dass in den ersten Fitbereich, der Fitter nicht mehr konvergiert, da die Datenlage anscheinend zu schlecht ist. Ab dem Fitbereich 8 - 25 funktioniert es wieder. Der niedrigste Zustand das D-Meson wird wieder sehr gut reproduziert. Auch der zuvor neu aufgetauchte Zustand wird wieder gefunden. Zusätzlich sieht man jetzt einen dritten Zustand, der zwischen den beiden liegt. Dieser könnte das  $D_0^*$  sein.

Man sieht also, dass für die Ergebnisse die Wahl der Fitbereiche von entscheidender Bedeutung ist.

## 4.5 Charmonium $6 \times 6$ Matrix: $J = 0$

Die  $6 \times 6$  Matrix wird dadurch gebildet, dass man alle Operatoren aus 4 nimmt, die die Eigenschaft  $C = +$  haben.

$P = - : \eta_c(1S), \eta_c(2S)$

$P = + : \chi_{c0}(1S)$

Es wurden einmal 2 und einmal 3 Energieniveaus aus den Daten extrahiert:

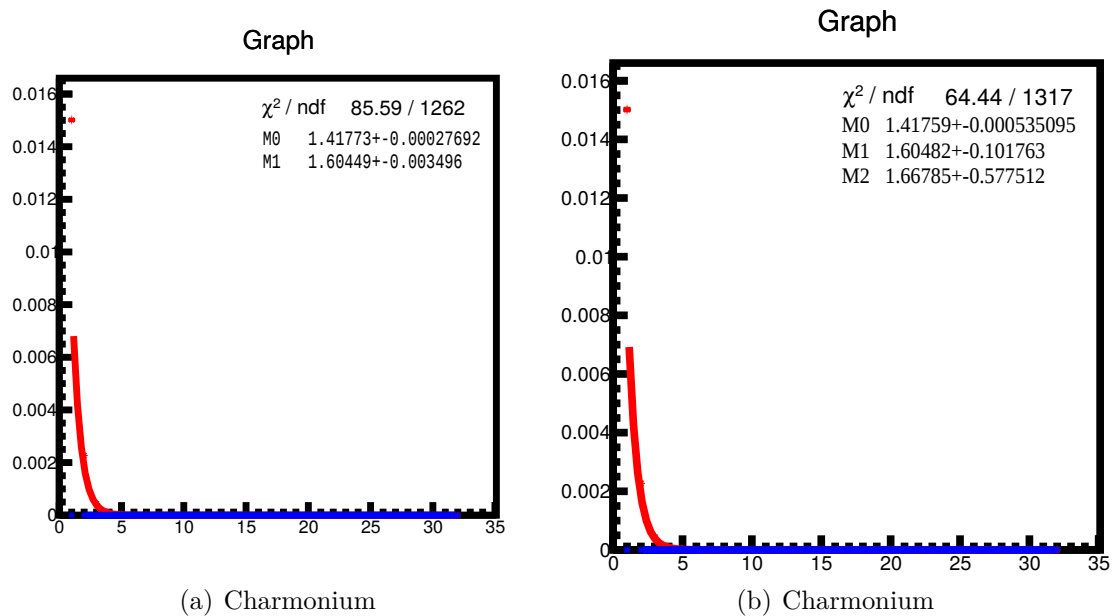


Abbildung 9:  $6 \times 6$  Matrizen

Im Vergleich dazu die ersten 4 Energieniveaus durch das GEP:

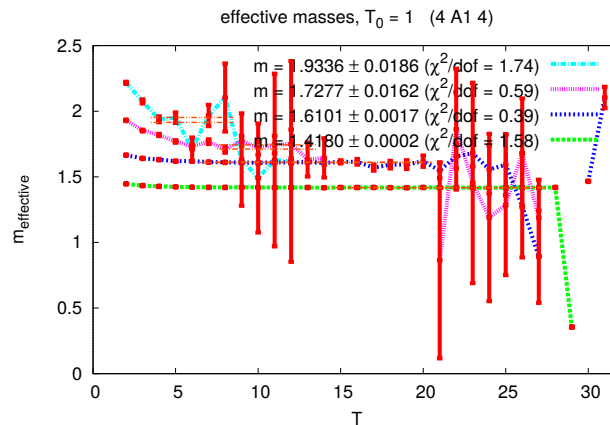


Abbildung 10:  $2 \times 2$  Matrix 2 Energien GEP

Die Ergebnisse sind in der folgenden Tabelle nochmals zusammengefasst:

	$M_0 \cdot a (\eta_c(1S))$	$M_1 \cdot a (\chi_{c0}(1S))$	$M_2 \cdot a (\eta_c(2S))$	$M_3 \cdot a$
MEF mit 2 Energien	$1,4177 \pm 0,0003$	$1,6045 \pm 0,0035$	-	-
MEF mit 3 Energien	$1,4176 \pm 0,0005$	$1,6048 \pm 0,1018$	$1,6668 \pm 0,5775$	-
GEP	$1,4180 \pm 0,0002$	$1,6101 \pm 0,0017$	$1,7277 \pm 0,0162$	$1,9336 \pm 0,0186$

Man kann erkennen, dass die Ergebnisse für das  $\eta_c(1S)$  und das  $\chi_{c0}(1S)$  übereinstimmen. Für das  $\eta_c(2S)$  gibt es eine kleine Abweichung von etwas mehr als 3 Prozent, wobei der dazugehörige Fehler bei fast 35 Prozent liegt. Für das 4. Energieniveau konnte ich bis jetzt keine Feineinstellungen finden um das Ergebnis zu reproduzieren.

## 5 Ausblick

Für die Zukunft geht es nun darum, das Programm so zu verbessern, dass man es auch zur Analyse der Struktur von Mesonen verwenden kann. Die Informationen zur Struktur der Mesonen steckt in den Amplituden der Exponentialfunktionen. In dieser Arbeit habe ich bewusst auf diese Analyse verzichtet, da mir deren Ergebnisse nicht vertrauenswürdig erschienen. Hier gilt es also nochmal einen genauen Blick auf das Programm zu werfen um diese Ergebnisse zu verbessern.

Außerdem soll das Programm die Möglichkeit bekommen, ganze Spalten, Zeilen und einzelne Operatoren aus dem Fit auszuschließen ohne jedesmal in dem Options-File jedes betroffene Matrixelement einzeln ansprechen zu müssen. Diese Feature würde die Handlichkeit des Programmes deutlich verbessern.

## 6 Fazit

In Anbetracht der in Kapitel 4 vorgestellten Ergebnisse ist deutlich geworden, dass der Ansatz durch den Multi-Exponential-Fit vielversprechend ist. Gerade für bereits bekannte Matrizen könnte es möglich sein, durch geschickte Wahl der Matrixelemente und Zeitfenster die Ergebnisse zu verbessern. Trotzdem muss man, wie man in Kapitel 4.4 und 4.5 sehen kann, höchst vorsichtig im Umgang mit dem Programm sein, da unterschiedliche Fitoptionen die Ergebnisse deutlich verändern können.

Für das Untersuchen völlig neuer Matrizen scheint mir das Programm jedoch aufgrund der in 3.6 geschilderten Probleme zu instabil zu sein.

## Literatur

- [ALPS15] ALEXANDROU, C. ; LEONTIOU, T. ; PAPANICOLAS, C. N. ; STILIARIS, E.: Novel analysis method for excited states in lattice QCD: The nucleon case. In: *Phys. Rev. D* 91 (2015), Nr. 1, S. 014506. <http://dx.doi.org/10.1103/PhysRevD.91.014506>. – DOI 10.1103/PhysRevD.91.014506
- [Ber80] BERKSON, Joseph: Minimum Chi-Square, not Maximum Likelihood. (1980)
- [Ber13] BERLIN, Joshua: *Meson spectroscopy using mixed action setups with twisted mass sea quarks*, Goethe Universität Frankfurt, Masterarbeit, 2013
- [BGHZ] BRUNIER, Jan H. ; GEER, Gerard van d. ; HARDER, Günter ; ZAGIER, Don: *The 1-2-3 of Modular Forms*. Springer
- [FH12] FODOR, Zoltan ; HOELBLING, Christian: Light Hadron Masses from Lattice QCD. In: *Rev. Mod. Phys.* 84 (2012), S. 449. <http://dx.doi.org/10.1103/RevModPhys.84.449>. – DOI 10.1103/RevModPhys.84.449
- [GL] GATTRINGER, Christof ; LANG, Christian B.: *Quantum Chromodynamics on the Lattice*. Springer
- [Kna07] KNAP, Michael: *Jackknife und Bootstrap*, TU Graz, Bakkalaureatsarbeit, 2007
- [KW15] KALINOWSKI, Martin ; WAGNER, Marc: Masses of  $D$  mesons,  $D_s$  mesons and charmonium states from twisted mass lattice QCD. (2015)
- [Lan08] LANG, C. B.: The Hadron Spectrum from Lattice QCD. In: *Prog. Part. Nucl. Phys.* 61 (2008), S. 35–49. <http://dx.doi.org/10.1016/j.pnpnp.2007.12.026>. – DOI 10.1016/j.pnpnp.2007.12.026
- [Pre14] PRELOVSEK, Sasa: Hadron Spectroscopy. In: *PoS LATTICE2014* (2014), S. 015
- [PS] PESKIN, Michael E. ; SCHROEDER, Daniel V.: *An Introduction to Quantum Field Theory*. Westview Press
- [PWL13] PHILIPSEN, Owe ; WAGNER, Marc ; LOTTINI, Stefano: Quantenfeldtheorie II Skript. (2013)
- [Ric12] RICHARD, Jean-Marc: An introduction to the quark model. (2012)
- [Rot] ROTHE, Heinz J.: *Lattice Gauge Theories An Introduction*. World Scientific
- [Ryd] RYDER, Lewis H.: *Quantum Field Theory*. Cambridge University Press
- [Wag15] WAGNER, Marc: *Berechnung von Massen, Zerfällen und Struktur von Hadronen mit Methoden der Gitter QCD*, Goethe Universität Frankfurt, Habilitationsschrift, 2015

- 
- [WDKW13] WAGNER, Marc ; DIEHL, Stefan ; KUSKE, Till ; WEBER, Johannes: An introduction to lattice hadron spectroscopy for students without quantum field theoretical background, 2013

## **Danksagung**

Zuallererst möchte Ich mich bei Marc Wagner für die Möglichkeit bedanken in diesem Bereich der Physik arbeiten zu können. Weiterhin gilt ihm und Joshua Berlin der Dank für eine immer motivierende und kompetente Unterstützung während der Bachelorarbeitsphase. Zusätzlich möchte ich Martin Kalinowski danken, der gerade gegen Ende der Arbeitsphase immer ein offenes Ohr für meine Fragen hatte. Ein weiterer Dank gilt Moritz Mälzer, der mir bei Problemen während des Programmierens immer zur Verfügung stand. Der letzte Dank gilt Andreas Kühn, meinem Bruder und meiner Mutter für das eifrige Korrekturlesen meiner Bachelorarbeit.

# Eidesstattliche Erklärung

## Eidesstattliche Erklärung zur Bachelorarbeit

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

*Unterschrift :*

*Ort, Datum :*