



Implementierung von U. Wolff's Γ -Methode in „PLASMA“

Institut für theoretische Physik
Goethe Universität Frankfurt

Bachelorarbeit
zur Erlangung des akademischen Grades
Bachelor of Science

Maximilian Märzc
geboren am 20.12.1993 in Frankfurt am Main

Erstprüfer	Prof. Dr. Owe Philipsen
Zweitprüfer	Prof. Dr. Marc Wagner
Betreuerin	Dr. Francesca Cuteri

Inhaltsverzeichnis

1	Einleitung	2
2	Theoretische Aspekte der Datenanalyse	4
2.1	Autokorrelation	4
2.2	Binning	5
2.3	Jackknife	6
2.4	Momente der statistischen Verteilung	8
2.5	Gauß-Test	10
3	Γ-Methode nach Ulli Wolff	11
3.1	Abgeleitete Größen	12
3.2	Fehler mit der Γ -Methode	14
3.3	Fehler der Fehler	16
3.4	Wahl des Summationsfensters	17
4	PLASMA	18
5	UNEW	20
5.1	Output	21
6	Implementierung	23
7	Ergebnisse der Programme	27
7.1	Single Chain Ergebnisse	28
7.2	Merged Chain vs Replica	31
7.3	Andere Observablen	32
7.4	Übereinstimmung schlägt fehl	32
8	Fazit	35
A	Anhang	36

1 Einleitung

Blickt man in die Vergangenheit der Physik, erkennt man, dass die Forscher schon immer davon fasziniert waren, immer kleinere Dinge zu erforschen und analysieren.

Angefangen mit den ersten Überlegungen zu Atomen, von denen man am Anfang noch dachte sie seien die kleinsten existierenden Teilchen, weiter über gewonnene Erkenntnisse zu deren Struktur und die damit eingeführten Elektronen, Protonen und Neutronen

Heute sind wir in der Lage den Aufbau der meisten uns bekannten Teilchen mithilfe des Standardmodells zu beschreiben, auch wenn dieses Modell noch nicht alle Aspekte umfasst liefert es doch immerhin Informationen über drei der vier bekannten Wechselwirkungen. Diese sind die elektromagnetische-, schwache, und starke Wechselwirkung. Einzig die Gravitation lässt sich mithilfe des Standardmodells nicht erklären. Dieses besteht aus den Quarks up, down, charm, strange, top und bottom. Hinzu kommen noch sechs Leptonen mit dem Elektron, dem Myon und dem Tau sowie deren Neutrinos. Diese 12 Teilchen sind in drei Generationen aufgeteilt. Desweiteren enthält das Standardmodell noch die Austauschteilchen der Wechselwirkung, die sogenannten Eichbosonen. Das Eichboson der elektromagnetischen Wechselwirkung ist das Photon. Die schwache Wechselwirkung hat drei Eichbosonen, nämlich das Z^0 , das W^+ und das W^- -Boson. Das Gluon ist das Austauschteilchen der starken Wechselwirkung. Diese Wechselwirkung lässt sich mithilfe der Theorie der Quantenchromodynamik beschreiben. Hierbei werden, wie der Name schon verrät, den Quarks sowie den Gluonen Farben zugewiesen, die sich innerhalb eines Teilchens zu Weiß „addieren“ müssen. Versucht man ein einzelnes Quark aus einem Teilchen zu entfernen wird die Energie so groß, dass es ausreicht, um ein neues Quark-Antiquark Pärchen zu erzeugen. Diese beiden Teilchen binden sich direkt an die anderen Teilchen und es entstehen zwei neue Teilchen. Es ist somit in der Natur unmöglich Quarks oder Gluonen ungebunden zu beobachten. Dieses Phänomen nennt man Confinement. Möglich wird es jedoch in einem Quark-Gluonen-Plasma, hier ist das Confinement aufgehoben und die Quarks und Gluonen befinden sich in einem quasi-freiem Zustand. Dies tritt aber nur bei extrem hoher Temperatur und hohem Druck auf, welche auf natürlichem Wege nicht zustandekommen. Forscher stellen diesen Zustand z.B. am Cern mithilfe des LHC's her. Da die meisten Berechnungen für Vielteilchensysteme analytisch nicht lösbar sind, hilft man sich hierbei mit Simulationen aus, welche die Lösung des Problems nähern.

Das Ziel dieser Bachelorarbeit ist es, das Open Source Tool UNEW in das PLASMA Programm der Arbeitsgruppe von Prof. Dr. Owe Philipsen zu implementieren. Der Grund hierfür liegt darin, die Fehler der statistischen Fehler der Analyse von Monte Carlo Simulationen zu verringern. Möglich wird dies mithilfe von Ulli Wolff's Γ -Methode.

Ich möchte zu Beginn auf theoretische Aspekte der Datenanalyse eingehen und kurz erklären, wie das PLASMA Tool zur Zeit damit umgeht.

Im Anschluss erläutere ich die Γ -Methode und deren Vorteile. Außerdem werde ich den Aufbau und die Funktion des UNEW Tools darlegen sowie die Implementierung in das PLASMA Tool. Am Ende werde ich noch Daten mit beiden Methoden auswerten und deren Ergebnisse vergleichen.

Die oben genannten Daten werden dabei in Simulationen erzeugt. Ziel dieser Simulationen ist es, Phasenübergänge zu einem Quark-Gluonen-Plasma in Quantengittern zu simulieren. Diese Simulationen sind Monte Carlo Simulationen. Mit ihnen lassen sich die analytisch nicht oder nur schwer lösbaren Probleme mithilfe der Wahrscheinlichkeitstheorie lösen. Innerhalb einer Simulation wird hierbei eine große Anzahl an Zufallsexperimenten wiederholt und die erzeugten Werte gespeichert. Eine solche Simulation ist immer für genau eine Konfiguration der äußeren Parameter zuständig. Es müssen also viele Simulationen durchgeführt werden, um Phasenübergänge beobachten zu können. Meistens wird dafür die Temperatur variiert.

Die Werte der Zufallsexperimente einer Simulation sind allerdings nur pseudozufällig. Dies liegt an der Tatsache, dass Computer keine absolut zufälligen Werte erzeugen können. Daraus folgt für die Werte, dass sie einer Korrelation unterliegen. Um diese Korrelation der Daten zu schätzen, werden Markov Ketten benutzt.

Mithilfe der Analyse des Mittelwertes und höheren statistischen Momenten lässt sich dann sagen, ob und wann ein Phasenübergang stattgefunden hat.

2 Theoretische Aspekte der Datenanalyse

Nachdem die pseudozufälligen Daten erzeugt wurden, lassen sich diese nun mithilfe von Markov Ketten analysieren.

Markov-Ketten bieten den Vorteil mit nur einem Teil der vorherigen Daten eine Vorhersage treffen zu können. Außerdem kann man mithilfe von Markov-Ketten die Autokorrelation der erzeugten Werte abschätzen, sowie den Fehler dieser Schätzung. In diesem Abschnitt möchte ich also auch auf die Fehleranalyse von Markov Ketten eingehen.

Wir gehen hier davon aus, dass die erzeugten Werte im Gleichgewicht sind. Es werden also nur Werte gespeichert, welche schon in einer Gleichverteilung sind bzw. diese erfüllen. Meistens werden hierfür zu Beginn der Erzeugung die Daten nicht gespeichert bzw. im Anschluss wieder verworfen.

2.1 Autokorrelation

Um den Erwartungswert einer Observablen f zu schätzen, benötigt man die Autokorrelation der Markov-Ketten. Diesem Abschnitt liegt Kapitel 4.1 aus [1] zugrunde.

Die zufälligen Messwerte sind hier die Schritte in der Markov-Kette

$$f_i = f_i(x_i), \quad i = 1, \dots, N. \quad (1)$$

Der Schätzwert von \hat{f} ist definiert als

$$\bar{f} = \frac{1}{N} \sum f_i. \quad (2)$$

Somit lässt sich die Autokorrelationsfunktion für die Observable \hat{f} definieren als

$$\hat{\Gamma}(t) = \hat{\Gamma}_{ij} = \langle (f_i - \langle f_i \rangle)(f_j - \langle f_j \rangle) \rangle = \langle f_i f_j \rangle - \langle f_i \rangle \langle f_j \rangle = \langle f_0 f_t \rangle - (\hat{f})^2 \quad (3)$$

mit folgender Notation $t = |i - j|$

Die Autokorrelation beschreibt also die Korrelation von Werten zu unterschiedlichen Zeiten und t ist hierbei die zeitliche Differenz der beiden Werte.

Für große t verhält sich diese Funktion wie folgt:

$$\hat{\Gamma}(t) \sim \exp\left(-\frac{t}{\tau_{exp}}\right) \text{ für } t \rightarrow \infty \quad (4)$$

wobei τ_{exp} exponentielle Autokorrelationszeit ist.

Die Varianz von f ist ein Sonderfall der Autokorrelation mit

$$\hat{\Gamma}(0) = \sigma^2(f) \quad (5)$$

also bei $t = 0$ entspricht die Autokorrelation genau der Varianz

$$\sigma^2(f) = \hat{f}^2 - (\hat{f})^2. \quad (6)$$

Die Varianz des Schätzwertes \bar{f} und die Autokorrelationsfunktion $\hat{\Gamma}(t)$ sind nun

über folgende Relation miteinander verknüpft

$$\begin{aligned}\sigma^2(\bar{f}) &= \langle (\bar{f} - \hat{f})^2 \rangle = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \langle (f_i - \hat{f})(f_j - \hat{f}) \rangle \\ &= \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \left[\langle f_i f_j \rangle - \hat{f}^2 \right] = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \Gamma_{ij}.\end{aligned}\quad (7)$$

Die beiden Summen in dem letzten Term lassen sich noch weiter vereinfachen, da $|i - j| = 0$ genau N Mal auftaucht und $|i - j| = t$ mit $1 \leq t \leq (N - 1)$ genau $2(N - t)$ Mal. Klammert man nun auch noch $\hat{\Gamma}(t)/N$ aus, folgt für die Gleichung

$$\sigma^2(\bar{f}) = \frac{\sigma^2(f)}{N} \left[1 + 2 \sum_{t=1}^{N-1} \left(1 - \frac{t}{N} \right) \hat{\Gamma}'(t) \right] \quad \text{mit } \hat{\Gamma}'(t) = \frac{\hat{\Gamma}(t)}{\hat{\Gamma}(0)}.\quad (8)$$

Hier fällt nun auf, dass die Varianz von \bar{f} für korrelierte Daten genau um den Faktor in Klammern größer als die Varianz von \bar{f} für unkorrelierte Daten ist. Dieser Faktor ist die integrierte Autokorrelationszeit

$$\tau_{int} = \left[1 + 2 \sum_{t=1}^{N-1} \left(1 - \frac{t}{N} \right) \hat{\Gamma}'(t) \right].\quad (9)$$

Da den meisten wissenschaftlichen Simulationen sehr viele Daten zugrunde liegen, sind wir nun also an dem Grenzwert $N \rightarrow \infty$ interessiert. Gl. (9) wird im Grenzwert $N \rightarrow \infty$ also zu:

$$\tau_{int} = 1 + 2 \sum_{t=1}^{\infty} \hat{\Gamma}'(t)\quad (10)$$

Betrachtet man für τ_{int} im Limes nun den Schätzwert wird man feststellen, dass dieser divergiert. Dies hat den Grund, dass jeder Schritt von einem Rauschen belastet ist, wohingegen das Signal nach Gl. (4) mit e^{-t} schwächer wird und schnell von dem Rauschen überlagert wird.

Um einen geeigneten Wert für $\bar{\tau}_{int}$ zu finden, gibt es nun verschiedene Möglichkeiten. Betrachten wir nun wieder den von t abhängigen Schätzwert für τ_{int}

$$\bar{\tau}_{int} = 1 + 2 \sum_{t'=1}^t \hat{\Gamma}'(t')\quad (11)$$

Man kann nun nach einem Bereich suchen, für den $\bar{\tau}_{int}$ nahezu unabhängig von t ist oder man benutzt die binning Methode, um sich dem Schätzwert zu nähern.

2.2 Binning

Unter Binning versteht man in der Datenanalyse das Zusammenfassen von Daten aus Intervallen zu jeweils einem Wert. Ein Folge von Zahlen wird also in gleichgroße Intervalle bzw. bins aufgeteilt und anschließend wird jedes Intervall

durch die Zahl ersetzt, die dieses Intervall ersetzen soll. Meistens wird hierfür der Mittelwert genommen.

Dadurch erzielt man eine Reduzierung der Daten. Bei einer Reduzierung verliert man natürlich auch einen Teil der Informationen. Bei geeigneter Größe der bins, also der Intervallbreite, können wir annehmen, dass die Daten nun nicht mehr korreliert sind.

Wir wollen nun das binning dazu verwenden, um die integrierte Autokorrelationszeit abzuschätzen. Dafür benutzen wir folgendes Verhältnis

$$\tau_{int} = \frac{\sigma^2(\bar{f})}{\sigma_{naive}^2(f)} \quad \text{mit} \quad \sigma_{naive}^2(\bar{f}) = \frac{\sigma^2(f)}{N}. \quad (12)$$

naive wurde hier benutzt, da hier einfach von unkorrelierten Daten ausgegangen wird.

Auch hier werden die Werte in den bins gemittelt

$$f_j^{N_b} = \frac{1}{N_b} \sum_{i=1+(j-1)N_b}^{jN_b} f_i \quad \text{für} \quad j = 1, \dots, N_{bs} \quad (13)$$

Hierbei ist N_b die Anzahl der Daten pro bin, N_{bs} die Anzahl der bins und N die Gesamtzahl der Daten.

Die integrierte Autokorrelationszeit lässt sich nun aus folgendem Verhältnis berechnen:

$$\tau_{int} = \lim_{N_b \rightarrow \infty} \tau_{int}^{N_b} \quad \text{mit} \quad \tau_{int}^{N_b} = \left(\frac{s_{\bar{f}^{N_b}}^2}{s_{\bar{f}}^2} \right) \quad (14)$$

also das Verhältnis der Schätzwerte der Varianzen der binned Daten zu allen Daten. Der Limes wird hier auch schon bei endlichen Zahlen erreicht.

Der Fehler von τ_{int} wird hier bestimmt durch den Fehler des Zählers, welcher deutlich größer ist, als der Fehler des Nenners in Gl. (14). Am besten wird das Ergebnis, wenn die bins gerade groß genug sind damit diese unkorreliert sind.

Auch dies Beschreibungen entsprechen Kapitel 4.1 in [1].

2.3 Jackknife

Die Jackknife Methode nach M. H. Quenouille und J. W. Tukey ist eine statistische Methode zur Abschätzung des Bias einer Stichprobe, wobei der Bias die Abweichung des Erwartungswertes zum wahren Wert beschreibt.

Folgende Beschreibungen sind nach Kapitel 2.7 aus [1]

Ist der Wert nicht Bias belastet, liefert die Jackknife-Methode den gewöhnlichen Fehler bzw. stimmt mit diesem überein. Ist dies nicht der Fall lässt sich der Bias und dessen Fehler mit dieser Methode deutlich besser abschätzen.

Mit den unkorrelierten Daten x_1, x_2, \dots, x_N ist

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (15)$$

für einen erwartungstreuen Erwartungswert $\hat{x} = \langle \bar{x} \rangle = \langle x \rangle$. Problematisch wird es erst, wenn man eine nicht lineare Funktion $\hat{f} = f(\hat{x})$ schätzen möchte.

Eine simple aber schlecht Schätzung für \hat{f} wäre

$$\bar{f}_{bad} = \frac{1}{N} \sum_{i=1}^N f_i \text{ mit } f_i = f(x_i). \quad (16)$$

Dies ist eine schlechte Schätzung, da der Bias in diesem Fall sehr groß ist.

$$\text{Bias}(\bar{f}_{bad}) = \hat{f} - \langle \bar{f}_{bad} \rangle = O(1) \quad (17)$$

Eine deutliche besser Schätzung für \hat{f} wäre

$$\bar{f} = f(\bar{x}). \quad (18)$$

Hiermit liegt der Bias nun deutlich besser bei

$$\text{Bias}(\bar{f}) = \hat{f} - \langle \bar{f} \rangle = \frac{a_1}{N} + \frac{a_2}{N^2} + O\left(\frac{1}{N^3}\right) \quad (19)$$

Da nun $f_i = f(x_i)$ kein guter Schätzwert von \hat{f} mehr ist, können wir auch nicht mehr die Standardgleichung

$$s^2(\bar{f}) = \frac{1}{N} s^2(f) = \frac{1}{N(N-1)} \sum_{i=1}^N (f_i - \bar{f})^2 \quad (20)$$

benutzen, um die Varianz abzuschätzen. Deshalb definieren wir nun den Jackknife Schätzer:

$$\bar{f}^J = \frac{1}{N} \sum_{i=1}^N f_i^J \text{ mit } f_i^J = f(x_i^J), \quad (21)$$

wobei gilt

$$x_i^J = \frac{1}{N-1} \sum_{k \neq i} x_k. \quad (22)$$

Wie man bei Gl. (22) erkennen kann, wird in der Summe immer ein Wert aus der Stichprobe weggelassen und darüber der Mittelwert gebildet. Mit all diesen gemittelten Stichproben lässt sich dann der Jackknife Schätzer berechnen.

Daraus folgt nun der Schätzwert der Varianz

$$s_J^2(\bar{f}^J) = \frac{N-1}{N} \sum_{i=1}^N (f_i^J - \bar{f}^J)^2. \quad (23)$$

Es lässt sich zeigen, dass man bei einem erwartungstreuen Fall bei der gewöhnlichen Varianz landet.

Im Falle einer Bias belasteten Gleichungen folgt für den Bias jedes Jackknife Schätzer \bar{f}^J

$$\text{Bias}(\bar{f}^J) = \hat{f} - \langle \bar{f}^J \rangle = \frac{a_1}{N-1} + \frac{a_2}{(N-1)^2} + O\left(\frac{1}{N^3}\right). \quad (24)$$

2.4 Momente der statistischen Verteilung

Bei statistischen Verteilungen gibt es neben dem Erwartungswert auch noch andere wichtige Parameter. Dabei sind vor allem die Varianz, Schiefe (Skewness) und Wölbung (Kurtosis) von großer Bedeutung. Diese Parameter liefern Informationen über die Verteilung.

Wie wir diese Parameter erhalten, werden wir im folgenden Abschnitt nach [3, Abschnitt 3.4] sehen.

Wir beginnen damit, das Moment der Ordnung n von X zu definieren. Wobei X eine Zufallsvariable ist und n eine natürliche Zahl. Das Moment ist also definiert als:

$$m_n = E(X^n) \quad (25)$$

Für $k = 1$ lässt sich nun leicht erkennen, dass dies einfach der Erwartungswert der Verteilung bzw. von X ist.

Für reelle Zufallsvariablen mit der Dichtefunktion $f(x)$ folgt:

$$m_n = \int_{-\infty}^{\infty} x^n f(x) dx \quad (26)$$

und das zentrale Moment, welches die Verteilung um den Erwartungswert beschreibt, mit

$$\mu_n = E((X - m_1)^n) \quad (27)$$

und

$$\mu_n = \int_{-\infty}^{\infty} (x - m_1)^n f(x) dx \quad (28)$$

Diese Momente lassen sich aus den momenterzeugenden Funktionen für die jeweiligen Momente ableiten. Die momenterzeugende Funktion für das Moment ist definiert als:

$$M(t) = \int_{-\infty}^{\infty} e^{t \cdot x} f(x) dx \quad (29)$$

Indem man nun die e -Funktion als Summe schreibt, lässt sich die Erzeugende umschreiben zu

$$M(t) = \int_{-\infty}^{\infty} \sum_{k=0}^{\infty} \frac{(tx)^k}{k!} f(x) dx = \sum_{k=0}^{\infty} \frac{t^k}{k!} \int_{-\infty}^{\infty} x^k f(x) dx = \sum_{k=0}^{\infty} \frac{t^k m_k}{k!} \quad (30)$$

Sieht man, dass gilt

$$m_n = M^{(n)}(0) \quad (31)$$

Also lässt sich das n -te Moment schreiben als n -te Ableitung der momenterzeugenden Funktion M an der Stelle $t = 0$.

Analog lässt sich dies für das zentrale Moment und dessen erzeugende Funktion wiederholen mit

$$C(t) = \int_{-\infty}^{\infty} e^{t \cdot (x - m_1)} f(x) dx \quad (32)$$

als Erzeugende für das zentrale Moment.

Also folgt wieder:

$$\mu_n = C^{(n)}(0) \quad (33)$$

Alternativ lässt sich eine statistische Verteilung auch mithilfe der Kumulanten κ_n beschreiben. Die Erzeugende ist über die momenterzeugende Funktion und den natürlichen Logarithmus als

$$K(t) = \ln(M(t)) \quad (34)$$

definiert.

Die Kumulanten κ_n lassen sich also wieder analog berechnen zu

$$\kappa_n = K^{(n)}(0). \quad (35)$$

Es lässt sich nun zeigen, dass m_n , μ_n und κ_n wie folgt zusammenhängen:

$$\begin{aligned} \mu_1 &= 0 & \kappa_1 &= m_1 \\ \mu_2 &= \kappa_2 & \kappa_2 &= m_2 - m_1^2 \\ \mu_3 &= \kappa_3 & \kappa_3 &= m_3 - 3m_2m_1 + 2m_1^3 \\ \mu_4 &= \kappa_4 + 3\kappa_2^2 & \kappa_4 &= m_4 - 4m_3m_1 - 3m_2^2 + 12m_2m_1^2 - 6m_1^4 \end{aligned}$$

Wie wir schon wissen, ist m_1 der Erwartungswert der Verteilung und die Varianz definiert über μ_2

$$\mu_2 = m_2 - m_1^2 \quad (36)$$

Für den Erwartungswert wird auch häufig μ verwendet sowie für die Varianz σ^2 .

Weiterhin fällt auf, dass μ_1 per Definition immer 0 ist. Fehlen also noch Skewness und Kurtosis.

Hierfür benötigen wir nun das normierte Moment B_n definiert über

$$B_n = \frac{\mu_n}{\mu_2^{\frac{n}{2}}}. \quad (37)$$

Die Skewness ist nun definiert als B_3

$$B_3 = \frac{\mu_3}{(\mu_2)^{3/2}} = \frac{m_3 - 3m_2m_1 + 2m_1^3}{(m_2 - m_1^2)^{3/2}}. \quad (38)$$

und die Kurtosis entspricht B_4

$$B_4 = \frac{\mu_4}{(\mu_2)^{4/2}} = \frac{m_4 - 4m_3m_1 - 3m_2^2 + 12m_2m_1^2 - 6m_1^4}{(m_2 - m_1^2)^2}. \quad (39)$$

Aber was genau sagt die Skewness über eine Verteilung aus?

Die Skewness gibt an wie und in welche Richtung die Verteilung schief ist. Schief ist eine Verteilung genau dann, wenn die Verteilung nicht symmetrisch ist.

Wenn die Verteilung also auf der linken Seite steiler ansteigt als auf der rechten Seite, nennt man die Verteilung rechtsschief und umgekehrt(rechts steiler als links) linksschief. Der Wertebereich der Skewness sind alle reellen Zahlen. Bei einer linksschiefen Verteilung ist die Skewness < 0 und bei einer rechtsschiefen Verteilung gerade < 0 .

Die Kurtosis beschreibt nun wie spitz eine Verteilung ist. Der Wertebereich der Kurtosis ist im Gegensatz zur Skewness nur im positiven Bereich. Ein hoher Wert bedeutet eine sehr spitze Verteilung, also wenige extrem häufige Ereignissen und sehr seltenen Ereignissen. Ist der Wert sehr nahe an der 0, handelt es sich um eine sehr flache Verteilung, bei der viele Ereignisse relative gleich häufig vorkommen.

Eine Normalverteilung vergleichsweise hat die Werte Skewness 0 und Kurtosis 3. Wobei die Skewness klar sein sollte, da es sich um eine symmetrische Verteilung handelt.

2.5 Gauß-Test

Um die unterschiedlich errechneten Erwartungswerte und deren Fehler auch vergleichen zu können, greifen wir auf den sogenannten Gauß-Test zurück, welcher auf den Seiten 60, 61 in [1] beschrieben wird. Angenommen wir haben die beiden Schätzwert \bar{x} und \bar{y} auf unterschiedliche Arten bei der selbe Normalerteilung bestimmt, wobei gilt, dass die Verteilung den Erwartungswert \hat{x} besitzt und die beiden Schätzwerte einer Varianz von σ_x^2 und σ_y^2 unterliegen. Für die Erwartungswerte gilt, da es die gleiche Verteilung ist, $\hat{x} = \hat{y}$. Da sich die Schätzwerte unterscheiden können, stellt sich die Frage, wie groß die Differenz

$$D = \bar{x} - \bar{y} \quad (40)$$

werden kann, bis man einen wirklichen Unterschied feststellen kann, welcher nicht statistischer Natur ist. Aus Gl. (40) folgt, dass D ebenfalls normalverteilt mit dem Erwartungswert

$$\hat{D} = 0 \quad (41)$$

sowie der Varianz

$$\sigma_D^2 = \sigma_x^2 + \sigma_y^2 \quad (42)$$

ist.

Weiterhin ist auch

$$d' = \frac{D}{\sigma_D} \quad (43)$$

normalverteilt mit der Varianz 1. Hierfür lässt sich mithilfe von

$$G(x) = \int_{-\infty}^x g(x') dx' = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x/\sigma} e^{-(x'')^2/2} dx'' = \frac{1}{2} + \frac{1}{2} \operatorname{erf} \left(\frac{x}{\sigma\sqrt{2}} \right) \quad (44)$$

wobei $g(x)$ die Wahrscheinlichkeitsdichte der Normalverteilung und erf die Fehlerfunktion

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-(x')^2} dx' \quad (45)$$

ist. Die Wahrscheinlichkeit für $|d'| \leq d$ finden zu

$$P = P(|d'| \leq d) = G_0(d) - G_0(-d) = 1 - 2G_0(-d) = \operatorname{erf} \left(\frac{d}{\sqrt{2}} \right). \quad (46)$$

Hiermit lässt sich nun die Wahrscheinlichkeit angeben, mit der die Differenz der beiden Schätzwerte zufällig ist, als

$$Q = 1 - P = 1 - \operatorname{erf} \left(\frac{d}{\sqrt{2}} \right). \quad (47)$$

Nun lässt sich hier eine Toleranzgrenze Q_{cut} festlegen und die beiden berechneten Schätzwerte unterscheiden sich für $Q \leq Q_{cut}$ signifikant. In der meisten Literatur wird der Wert für $Q_{cut} = 0.05$ als Standardwert gewählt, wobei der Wert selbst frei gewählt werden kann.

3 Γ -Methode nach Ulli Wolff

Wir haben die Grundlagen der theoretischen Datenanalyse für Markov-Ketten kennen gelernt und beginnen nun mit der Γ -Methode. Bei dieser Methode werden die Autokorrelationsfunktion und -zeiten explizit bestimmt.

Auch hierbei nehmen wir wieder an, dass die Markov-Ketten schon im Gleichgewicht sind.

Dieser Abschnitt beschreibt die Γ -Methode Anhand von [4].

Autokorrelationsfunktion $\Gamma_{\alpha\beta}$ identisch zu Gl. (3) mit anderen Bezeichnungen
Wir beginnen auch in diesem Abschnitt wieder mit der Autokorrelationsfunktion

$$\Gamma_{\alpha\beta}(n) = \langle (a_\alpha^i - A_\alpha)(a_\beta^{i+n} - A_\beta) \rangle, \quad (48)$$

wobei a_α^i eine primäre Observable und $A_\alpha = \langle a_\alpha \rangle$ deren Erwartungswert ist. Der Index i beschreibt hier aufeinander folgende Schätzwerte der Observablen mit $i = 1, \dots, N$, welche durch eine update Prozedur voneinander getrennt sind. Wir benutzen nun die Simulation um Konfigurationen ϕ_i , welche normalisiert Boltzmannverteilt sind ($P(\phi)$), zu erzeugen. Die update Prozedur wird nun beschrieben durch eine Übergangswahrscheinlichkeit $W(\phi \rightarrow \phi')$ und die Messwerte durch $a_\alpha^i = O_\alpha(\phi_i)$.

Dann lässt sich die Autokorrelationsfunktion schreiben als

$$\Gamma_{\alpha\beta}(n) = \sum_{\phi} \sum_{\phi'} P(\phi) W^n(\phi \rightarrow \phi') (O_\alpha(\phi) - A_\alpha)(O_\beta(\phi') - A_\beta) \quad (49)$$

über die Konfigurationen, wobei W^n die Übergangswahrscheinlichkeit nach n Ausführungen der update Prozedur ist. Es ist weiterhin nützlich eine Symmetrie anzugeben

$$\Gamma_{\alpha\beta}(-n) = \Gamma_{\beta\alpha}(n) \quad (50)$$

Es lässt sich noch eine weitere Indize für die replica $r = 1, \dots, R$ in die obigen Gleichungen einfügen. Die Observable ist nun als $a_\alpha^{i,r}$. Dieser beschreibt die unabhängigen Simulationen für die gleiche Observable a_α^i . Gl. (48) wird nun also zu

$$\langle (a_\alpha^{i,r} - A_\alpha)(a_\beta^{j,s} - A_\beta) \rangle = \delta_{rs} \Gamma_{\alpha\beta}(j - i). \quad (51)$$

Bei nur einer einzigen Wiederholung der Simulation ist $R = 1$. Wenn mehrere Replicas benutzt werden, bedeutet das für den Index $i = 1, \dots, N_r$. Dies bedingt:

$$N = \sum_{r=1}^R N_r \quad (52)$$

Der Durchschnitt für jedes replica ist also

$$\bar{a}_\alpha^r = \frac{1}{N_r} \sum_{i=1}^{N_r} a_\alpha^{i,r} \quad (53)$$

und der Schätzwert für A_α lautet dann

$$\bar{a}_\alpha = \frac{1}{N} \sum_{r=1}^R N_r \bar{a}_\alpha^r. \quad (54)$$

Die Abweichungen zwischen Mittelwert und dem Wert selbst, sowohl für replica als auch gesamt, sind also

$$\bar{\delta}_\alpha^r = \bar{a}_\alpha^r - A_\alpha, \quad \bar{\bar{\delta}}_\alpha = \bar{\bar{a}}_\alpha - A_\alpha. \quad (55)$$

Wir gehen nun davon aus, dass der Schätzwert erwartungstreu ist, also dem Erwartungswert entspricht, bzw. der Bias 0 ist.

$$\langle \bar{\delta}_\alpha^r \rangle = 0 = \langle \bar{\bar{\delta}}_\alpha \rangle \quad (56)$$

Auch wenn $a_\alpha^{i,r}$ nicht selbst einer Gaußverteilung entspricht, wird \bar{a}_α^r durch den zentralen Grenzwertsatz normalverteilt sein. Deshalb können wir nun eine Normalverteilung annehmen, welche durch folgende Kovarianzmatrix beschrieben wird:

$$\langle \bar{\delta}_\alpha^r \bar{\delta}_\beta^s \rangle = \frac{1}{N_r^2} \sum_{i,j=1}^{N_r} \Gamma_{\alpha\beta}(i-j) \delta_{rs} = \frac{1}{N_r} C_{\alpha\beta} \delta_{rs} \times (1 + O(\tau/N_r)) \quad (57)$$

mit

$$C_{\alpha\beta} = \sum_{t=-\infty}^{\infty} \Gamma_{\alpha\beta}(t) \quad (58)$$

Wie man sehen kann, hängt diese nicht mehr von der Laufzeit t ab. Es wird weiterhin angenommen, dass das endliche τ hier den exponentiellen Abfall von $\Gamma_{\alpha\beta}$ mit $|t|$ beschreibt. Eine weitere Bedingung, um eine Fehlerbestimmung möglich zu machen liegt in N_r . Dies ist nur möglich wenn gilt $N_r \gg \tau$.

Die Kovarianzmatrix lässt sich nun für $\bar{\bar{a}}_\alpha$ schreiben als

$$\langle \bar{\bar{\delta}}_\alpha \bar{\bar{\delta}}_\beta \rangle = \frac{1}{N} C_{\alpha\beta} \times (1 + O(R\tau/N)). \quad (59)$$

Einfach gesagt weicht $\bar{\bar{a}}_\alpha$ um einen Fehler der Größenordnung $\frac{1}{\sqrt{N}}$ von A_α ab. Außerdem lässt sich erkennen, dass die Fehleranalyse im Wesentlichen von einer Bestimmung von $C_{\alpha\beta}$ abhängt, da $C_{\alpha\beta}$ alle Autokorrelationseffekte enthält.

3.1 Abgeleitete Größen

Für abgeleitete Größen bzw. Werte F definieren wir analog zu den primären Observablen

$$\bar{\bar{F}} = f(\bar{\bar{a}}_\alpha) \quad (60)$$

und

$$\bar{F} = \frac{1}{N} \sum_{r=1}^R N_r f(\bar{a}_\alpha^r) \quad (61)$$

Führt man für den Schätzwert jetzt eine Tayler-Entwicklung für f unter der Annahme, dass die primäre Observable genau genug ist, durch, erhalten wir

$$\bar{\bar{F}} = F + \sum_{\alpha} f_{\alpha} \bar{\bar{\delta}}_{\alpha} + \frac{1}{2} \sum_{\alpha\beta} f_{\alpha\beta} \bar{\bar{\delta}}_{\alpha} \bar{\bar{\delta}}_{\beta} + \dots \quad (62)$$

wobei gilt

$$f_{\alpha} = \frac{\partial f}{\partial A_{\alpha}}, \quad f_{\alpha\beta} = \frac{\partial^2 f}{\partial A_{\alpha} \partial A_{\beta}} \quad (63)$$

daraus lässt sich schließen, dass $\bar{\bar{F}}$ folgendem Bias unterliegt, es sei denn f ist linear

$$\langle \bar{\bar{F}} - F \rangle \approx \frac{1}{2N} \sum_{\alpha\beta} f_{\alpha\beta} C_{\alpha\beta} \approx \frac{1}{R} \langle \bar{F} - F \rangle, \quad (64)$$

wobei wir folgende Notation benutzt haben

$$C_{\alpha\beta} = \sum_{t=-\infty}^{\infty} \Gamma_{\alpha\beta}(t). \quad (65)$$

Dieser Bias ist allerdings bei großen N vernachlässigbar klein verglichen mit den statistischen Fehlern der Simulation.

Durch mehrere Replicas lässt sich der Bias noch weiter verkleinern, mithilfe von

$$\langle \bar{\bar{F}} - F \rangle \approx \frac{1}{R-1} \langle \bar{F} - \bar{F} \rangle \quad (66)$$

lässt sich nun $\bar{\bar{F}}$ ersetzt zu

$$\bar{\bar{F}} \rightarrow \frac{R\bar{F} - \bar{F}}{R-1}. \quad (67)$$

Betrachtet man nun die Varianz, erfährt man den Fehler zwischen Schätzwert und wahren Wert:

$$\sigma_F^2 = \langle (\bar{\bar{F}} - F)^2 \rangle \approx \frac{1}{N} \sum_{\alpha\beta} f_{\alpha} f_{\beta} C_{\alpha\beta} \equiv \frac{C_F}{N} \quad (68)$$

mit

$$C_F = \sum_{\alpha\beta} f_{\alpha} f_{\beta} C_{\alpha\beta} \quad (69)$$

Die Varianz lässt sich nun mithilfe der effektiven Varianz von F , also eine naive bzw. ohne Autokorrelation berechnete Varianz,

$$v_F = \sum_{\alpha\beta} f_{\alpha} f_{\beta} \Gamma_{\alpha\beta}(0) \quad (70)$$

und der integrierten Autokorrelationszeit

$$\tau_{int,F} = \frac{1}{2v_F} \sum_{t=-\infty}^{\infty} \sum_{\alpha\beta} f_{\alpha} f_{\beta} \Gamma_{\alpha\beta}(t) \quad (71)$$

umschreiben zu:

$$\sigma_F^2 = \frac{2\tau_{int,F}}{N} v_F \quad (72)$$

Wie wir sehen können, taucht nun schon eine wichtige Größe für die Analyse auf: die integrierte Autokorrelationszeit. Diese gibt eine erste Schätzung für die Autokorrelationszeit an.

Betrachten wir nun die Gl. (72) für die Varianz in zwei Extremfällen.

Wenn gar keine Autokorrelation auftritt, erhalten wir $2\tau_{int,F} = 1$ und Gl. (72) wird einfach zu dem Standardfehler.

Der zweite Extremfall tritt ein, wenn die Autokorrelationsfunktion ein rein exponentielles Verhalten aufweist

$$\Gamma_{\alpha\beta}(t) \sim e^{-\frac{|t|}{\tau}} \quad (73)$$

folgt für $\tau_{int,F}$

$$\tau_{int,F} = \tau + O(\tau^{-1}) \quad (74)$$

Man kann Gl. (72) auch dahin gehend interpretieren, dass nur $N/(2\tau_{int,F})$ Werte nicht autokorreliert sind und die statistischen Fehler verringern.

Die Schätzwerte der Replicas $f(\bar{a}_\alpha^r)$ sollten normalverteilt sein und deren Varianzen sind

$$\langle (f(\bar{a}_\alpha^r) - F)^2 \rangle = \frac{C_F}{N_r} \quad (75)$$

Hierüber lässt sich nun ein Verteilungstest bzw χ^2 -Test der Replica Schätzwerte durchführen. Hierfür muss man

$$\chi^2(K) = \sum_r \frac{(f(\bar{a}_\alpha^r) - K)^2}{C_F/N_r} \quad (76)$$

in Abhängigkeit der Konstanten K minimieren.

Das Minimum wird man dabei bei

$$K = \frac{1}{N} \sum_{r=1}^R N_r f(\bar{a}_\alpha^r) = \bar{F} \quad (77)$$

finden. Die Wahrscheinlichkeit ein Minimum mit $\chi^2 \geq x$ zu finden liegt dabei bei

$$Q = 1 - P\left(\frac{x}{2}, \frac{(R-1)}{2}\right). \quad (78)$$

Wobei P die unvollständige Gammafunktion

$$P(U, n) = \frac{1}{\Gamma(n)} \int_0^U du u^{n-1} \exp(-u) \quad (79)$$

ist.

3.2 Fehler mit der Γ -Methode

Nachdem wir nun die wesentlichen Dinge für die Fehlerbetrachtung definiert und kennengelernt haben, fangen wir nun mit der eigentlichen Γ -Methode an.

Mithilfe dieser Methode schätzen wir den Fehler und die Autokorrelation direkt ab. Dies steht im direkten Gegensatz zur Binning-Methode (siehe §2.2).

Wir definieren zuerst den Schätzwert der Autokorrelationsfunktion

$$\bar{\bar{\Gamma}}_{\alpha\beta}(t) = \frac{1}{N - Rt} \sum_{r=1}^R \sum_{i=1}^{N_r-t} (a_\alpha^{i,r} - \bar{a}_\alpha)(a_\beta^{i+t,r} - \bar{a}_\beta), \quad (80)$$

wobei gilt $0 \leq t \ll N_r \quad \forall r$.

Dies wurde so gewählt, um unnötiges Rauschen zu vermeiden. Außerdem wird

die Korrelation erst für jedes Replica und am Ende über alle Replica gemittelt. Die Autokorrelationsfunktion für eine abgeleitete Größe F lautet:

$$\bar{\bar{\Gamma}}_F(t) = \sum_{\alpha\beta} \bar{f}_\alpha \bar{f}_\beta \bar{\bar{\Gamma}}_{\alpha\beta}(t) \quad (81)$$

mit

$$\bar{f}_\alpha = \frac{\partial f}{\partial \bar{a}_\alpha} \quad (82)$$

also wie bei der Taylorentwicklung nur an den Stellen $\bar{a}_1, \bar{a}_2, \dots$ ausgewertet. Dies fügt einen relativen Fehler der Ordnung $1/N$ hinzu.

Die Schätzwerte für v_F und C_F lauten

$$\bar{v}_F = \bar{\bar{\Gamma}}_F(0) \quad (83)$$

und

$$\bar{\bar{C}}_F(W) = \left[\bar{\bar{\Gamma}}_F(0) + 2 \sum_{t=1}^W \bar{\bar{\Gamma}}_F(t) \right]. \quad (84)$$

Der Bias, der durch den Abbruch der Summe in der Autokorrelationsfunktion entsteht, lautet wie folgt

$$\frac{\langle \bar{\bar{C}}_F(W) \rangle - C_F}{C_F} \sim -e^{-\frac{W}{\tau}}. \quad (85)$$

Hierbei steht das W für ein Fenster in dem summiert wird. Dieses Fenster muss gut gewählt werden, da es großen Einfluss auf die Auswertung hat, jedoch enthält es auch den oben genannten systematischen Fehler.

Zum einen muss es sehr groß im Vergleich zur Abklingzeit τ sein, um einen kleinen systematischen Fehler zu ermöglichen. Andererseits darf es nicht zu groß gewählt werden, um nicht ein zu großes Rauschen ohne wirkliche Daten einzubringen.

Oft tritt der Fall ein, dass man zwar viele primären Observablen A_α hat, jedoch nur wenigen Funktionen bestimmen möchte. In diesem Fall ist es meist von Vorteil, nicht die gesamte Autokorrelationsmatrix $\bar{\bar{\Gamma}}_{\alpha\beta}(t)$ zu bestimmen, sondern direkt \bar{a}_α und \bar{f}_α sowie die gemessenen Daten auf

$$a_\alpha^{i,r} \rightarrow a_f^{i,r} = \sum_{\alpha} \bar{f}_\alpha a_\alpha^{i,r} \quad (86)$$

zu projizieren und die Autokorrelation direkt damit zu berechnen.

Außerdem kann es natürlich bei der Berechnung der Ableitung von f zu Problem kommen. In diesem Fall bleibt nur die numerische Berechnung der Ableitung

$$\bar{f}_\alpha \approx \frac{1}{2h_\alpha} [f(\bar{a}_1, \dots, \bar{a}_\alpha + h_\alpha, \dots) - f(\bar{a}_1, \dots, \bar{a}_\alpha - h_\alpha, \dots)] \quad (87)$$

mit

$$h_\alpha = \sqrt{\frac{\bar{\bar{\Gamma}}_{\alpha\alpha}(0)}{N}}. \quad (88)$$

Auch hier erhalten wir wieder einen Fehler von der Ordnung $O(h_\alpha^2) \propto \frac{1}{N}$, welcher wieder vernachlässigbar ist.

3.3 Fehler der Fehler

Bei Simulationen und statistischen Auswertungen, bei denen die Fehler numerisch bestimmt wurden sind eben diese Fehler ebenfalls fehlerbehaftet. Zu wissen, wie groß dieser Fehler des Fehlers ist, ist vor allem dann wichtig, wenn man eine theoretische Vorhersage mit einer 2.5σ Diskrepanz machen möchte, aber σ selbst stark fehlerbehaftet ist.

Um eben den Fehler des Fehlers zu bestimmen, benutzen wir die Madras-Sokal Formel. Nach [4, Anhang A] erhalten wir mit eben dieser Gleichung folgende Näherungen

$$\langle (\bar{C}_F(W) - C_F)^2 \rangle \approx \frac{2(2W+1)}{N} C_F^2 \quad (89)$$

und für

$$\bar{\tau}_{int,F}(W) = \frac{\bar{C}_F(W)}{2\bar{v}_F} \quad (90)$$

erhalten wir

$$\langle (\bar{\tau}_{int,F}(W) - \tau_{int,F})^2 \rangle \approx \frac{4}{N} \left(W + \frac{1}{2} - \tau_{int,F} \right) \tau_{int,F}^2. \quad (91)$$

Wobei W wieder das Summationsfenster ist. Wie dieses genau gewählt wird, wird im nächsten Abschnitt beschrieben. Der optimale Wert für W ist natürlich jener, welchen die Summe der Fehler minimiert. In Abhängigkeit von W lässt sich der relative Fehler als

$$\frac{\delta_{tot}(\bar{\sigma}_F)}{\bar{\sigma}_F} \approx \frac{1}{2} \min_W \left(\exp(-W/\tau) + 2\sqrt{W/N} \right) \quad (92)$$

beschreiben. Für die gesamte Fehlerschätzung erhalten wir nun

$$\bar{\sigma}^2 = \frac{\bar{C}(W)}{N}. \quad (93)$$

Für den Fehler des Fehlers folgt weiterhin mithilfe von Gl. (89)

$$\delta_{stat}(\bar{\sigma}_F) = \sqrt{\frac{W+1/2}{N}}. \quad (94)$$

Mit Gl. (93) lässt sich eine gute Näherung mit

$$W = \tau \ln(N/\tau)/2 \quad (95)$$

machen. Dies ist hilfreich um zu sehen, dass sich der systematische Fehler von Gl. (85) bei sehr großen N vernachlässigbar klein wird.

$$\frac{\delta_{sys}(\bar{\sigma}_F)}{\delta_{stat}(\bar{\sigma}_F)} = \frac{\tau}{2W} \approx \frac{1}{\ln(N/\tau)} \quad (96)$$

3.4 Wahl des Summationsfensters

Wie zuvor schon erwähnt, werden wir uns jetzt mit der Wahl des Summationsfensters W beschäftigen. Dieses Fenster spielt eine ähnlich große Rolle, wie die Größe der bins in der Binning Methode.

Nach [4, Abschnitt 3.3] nehmen wir an, dass gilt

$$\tau \sim S \cdot \tau_{int,F}. \quad (97)$$

S ist hierbei ein einfacher Faktor.

Löst man nun

$$2\bar{\tau}_{int,F}(W) = \sum_{t=-\infty}^{\infty} \exp\left(-\frac{S|t|}{\bar{\tau}(W)}\right) \quad (98)$$

nach $\bar{\tau}$, erhält man

$$\frac{S}{\bar{\tau}(W)} = \ln\left(\frac{2\bar{\tau}_{int,F}(W) + 1}{2\bar{\tau}_{int,F}(W) - 1}\right). \quad (99)$$

Für $\bar{\tau}_{int,F} \leq \frac{1}{2}$ setzen wir $\bar{\tau}(W)$ auf einen Wert knapp über 0.

Es lässt sich nun Gl. (92), welche aus der Summe von systematischen und statistischen Fehler besteht, in Abhängigkeit von W minimieren, um W_{opt} zu erhalten. Die Ableitung nach W dieser Gleichung ist

$$g(W) = \exp\left(-\frac{W}{\bar{\tau}(W)}\right) - \frac{\bar{\tau}(W)}{\sqrt{WN}}. \quad (100)$$

Um W_{opt} zu bestimmen, wird $g(W)$ nun nach und nach für $W = 1, 2, \dots$ berechnet. Der erste Wert, bei dem W negativ ist, wird dann als W_{opt} festgelegt. Der Grund dafür liegt in dem Wechsel des Vorzeichens, da dies ein Extrema markiert. Schlägt diese Bestimmung von W_{opt} fehl, also existiert kein Vorzeichenwechsel bis ν

$$\nu = \min_r N_r/2 \quad (101)$$

setzen wir $W = \nu$. Die Autokorrelationsfunktion in Gl. (81) wird also berechnet über

$$t = t_{max} = 2\min(W_{opt}, \nu) \quad (102)$$

Meistens liegt der Faktor S hierbei in einem Bereich $S \in [1, 2]$. Ist er zu groß, führt er zu größeren Fehler der Fehler. Außerdem sollte der Bereich um S einigermaßen stabil sein, da wir uns an einem Extrema befinden. Ist dies nicht gegeben, lässt sich S noch verändert, um dies zu erreichen.

4 PLASMA

Mit PLASMA ist hier und im Folgenden das LQCD_SimulationManagement-Utilities.git der Arbeitsgruppe von Prof. Philipsen gemeint. Die Abkürzung steht hierbei für:

Python Library for Automatic Simulations Management and Analysis

Diese Bezeichnung wurde von Mitgliedern der Arbeitsgruppe vorgeschlagen, um eine besser und einfachere Bezeichnung für das Programm zu haben.

Die Datenanalyse lässt sich mit der Datei ImagMuAnalysis.py starten. Das Programm sollte man dabei in dem Verzeichnis der gespeicherten Daten starten. Dies hat den Grund, dass da Programm viele Informationen aus dem Pfad des Verzeichnisses herauslesen kann. Die Parameter, die ausgelesen werden können sind:

N_f	Die Anzahl der Flavors
μ_i	Der Wert des imaginären chemischen Potentials
κ	Der Wert des Hopping Parameters
N_τ	Der Wert der zeitlichen Ausdehnung des Gitters
N_σ	Der Wert der räumlichen Ausdehnung des Gitters

Beim Starten des Programms übergibt man entweder direkt oder in einer Datei die β -Werte, welche analysiert werden sollen innerhalb des Verzeichnisses.

Es können beim Ausführen des Programms verschieden Observablen analysiert werden. Diese sind:

AccRate also acceptance rate bzw. Akzeptanzrate. Diese beschreibt das Verhältnis von neuen akzeptierten Konfigurationen zur Gesamtzahl der Monte-Carlo Schritten und gibt Informationen darüber, wie schnell die Simulation den Phasenraum des Systems darstellt. Normalerweise liegt der Wert bei $\gtrsim 75\%$ nach Seite 80 in [3].

Plaq, also plaquette bzw. Plakette. Dies ist die Größe für die kleinste geschlossene Schleife.

Poly, also polyakov loop quantities sind allgemeiner und es gibt viele Möglichkeiten hierfür. Unter anderem den real-, imaginär-Teil oder auch den Betrag.

Die Daten für diese verschiedenen Observablen wurden bei den Simulationen erzeugt und in einer Datei gespeichert. PLASMA analysiert diese Daten nur und je nach gewählter Observablen werden andere Spalten aus der Datei gelesen.

In den Beispieldaten, die ich erhalten habe, wurde das Dateiformat „cl2qcd“ verwendet, bei denen die Anordnung der hier relevanten Werte in Tabelle 1 dargestellt werden. Für jede Observable werden die Zufallszahlen vom Programm

Observable	# Spalte
plaq	1
poly_re	4
poly_im	5
accRate	9

Tabelle 1: Relevante Spalten in den Dateien

eingelassen und anschließend die verschiedene Größen für jede Verteilung berechnet. Diese sind: Erwartungswert, Varianz, Skewness und Kurtosis.

Aber auch diese lassen sich, wenn erwünscht, einschränken und nur bestimmte Werte ermitteln. Das gleiche trifft auf die integrierte Autokorrelationszeit zu.

Wie analysiert PLASMA nun aber die Daten?

Es wird hierfür eine hybride Analysemethode verwendet. Das bedeutet, dass τ_{int} bereits über die Γ -Methode berechnet wird, allerdings keine anderen Werte. Sobald τ_{int} berechnet ist, werden die Werte benutzt, um zu die Größe von Bins zu berechnen. Bei der Berechnung von τ_{int} ist vor allem wichtig, dass mithilfe von Ulli Wolff's Γ -Methode nur die integrierte Autokorrelationszeiten der zentralen Momente berechnet werden. Um nun die Bingröße für die normierten Momente berechnen zu können, wird für Zähler und Nenner der gleiche Wert für τ_{int} gewählt. Dieser Wert für die normierten Momente ist das Maximum aus dem zweiten zentralen Moment(Nenner) und dem n-ten zentralen Moment(Zähler). Diese Ergebnisse sowie die Verteilung selbst wird im Anschluss an ein externes C++ Programm übergeben, welches die Daten binned und mithilfe der Jackknife-Methode die Analyse für die Momente der Verteilung berechnet.

5 UNEW

Das UNEW Programm [2] soll die statistischen Fehler schätzen und die Autokorrelation für Sequenzen nach der beschriebenen Γ -Methode berechnen. Dies ist vor allem bei den Monte Carlo Simulationen sehr wichtig.

Das Programm bietet dabei eine Vielzahl an Möglichkeiten der Datenanalyse und Auswertung, auch im Bereich der graphischen Auswertung.

In den nächsten Abschnitten möchte ich die Möglichkeiten des Programms sowie dessen Benutzung erklären.

Die Benutzung des Programms erfolgt über das Terminal. Es lässt sich entweder direkt die Pythondatei des Programms starten oder nach der Installation des Programms wird ein eigener Befehl zum Starten eingeführt. Dieser lautet simpel „unew“.

Bei UNEW lassen sich alle Angaben entweder direkt im Terminal machen oder man lädt eine yaml-Datei, in der alle Parameter gespeichert sind. Der Umfang der Möglichkeiten ist für beide Varianten gleich. Hier beschreibe ich vor allem die Vorgehensweise im Terminal. Es lassen sich mithilfe verschiedener Commandlineoptionen auswählen, welche Daten analysiert werden sollen und wie. Bei der Datenauswahl gibt es zwei verschiedenen Möglichkeiten vorzugehen. Die erste Möglichkeit besteht darin, jede einzelne Datei direkt in der Commandline anzugeben, wobei eine relative Angabe ausreichend ist. Bei der zweiten Möglichkeit wird nur ein Verzeichnis angegeben und im Anschluss kann noch ein Suchmuster eingegeben werden, um die Dateiauswahl in dem angegebenen Verzeichnis einzuschränken. Hierbei lassen sich dieselben Suchmuster wie in der Bash selbst verwenden (also z.B. mit „*.dat“ alle Dateien die auf „.dat“ enden).

Es lassen sich weiterhin auch die Spalten der ausgewählten Dateien auswählen sowie die Dateien in eine beliebige Anzahl an Replica spalten.

Eine weitere Möglichkeit ist die Verwendung von Funktionen, die auf die Verteilungen angewendet werden sollen. Hierfür muss der Name eines Pythonmoduls sowie der Funktionsname der gewünschten Funktion/en übergeben werden. Falls die Funktionen noch von anderen Werten abhängig sind, lassen diese sich natürlich auch in der Commandline festlegen. UNEW ist weiterhin in der Lage, Plots zu erzeugen und zu speichern. Ohne Angaben werden die Plots nur angezeigt und nicht gespeichert. Das lässt sich entweder ganz deaktivieren oder man verbirgt diese und speichert sie.

Die letzte Möglichkeit besteht darin, das Verhältnis $S = \tau/\tau_{int}$ zu schätzen. Der Standardwert liegt hier bei $S = 1,5$.

5.1 Output

Die berechneten Wert und deren Fehler werden für jeden Observable im Terminal ausgegeben. Das Programm berechnet die in Tabelle 2:

value	erwartungstreue Erwartungswert von F
error	Fehler von <i>value</i> nach Gl. (68) mit $W = W_{opt}$
error of error	statistischer Fehler von <i>error</i> mit $W = W_{opt}$
naive error	Fehler unter der Annahme, dass keine Autokorrelation auftritt
variance	Varianz
tau_int	integrierte Autokorrelationszeit mit $W = W_{opt}$
tau_int error	Fehler von <i>tau_int</i>
W_opt	optimaler Wert für W_{opt} für das Summationsfenster W nach Gl. (100)
t_max	maximaler Wert für t bis zu welchem Wert die Autokorrelationsfunktion $\Gamma_F(t)$ berechnet wird. Gegeben durch $2\min(W_{opt}, \nu)$ (Gl. (102))
Q_val	repräsentiert den Q -Wert für das Histogramm der Replica-Distributionen nach Gl. (78)

Tabelle 2: Ausgabewerte UNEW

In Listing 1 ist eine Beispielausgabe von UNEW zu sehen wie sie für eine primäre Observable im Terminal aussieht.

```

### RESULTS ###

Results for primary observable n. 1:
  value: 4.889976900284044e-01
  error: 9.018639671760418e-04
error of error: 1.085720508110510e-04
  naive error: 4.405410682994761e-05
  variance: 1.358535030009105e-04
  tau_int: 2.083367458183251e+02
tau_int error: 4.471552676425301e+01
  W_opt: 1014
  t_max: 2028
  Q_val: 0.000000000000000e+00

```

Listing 1: Beispiel einer UNEW-Ausgabe im Terminal

Wie schon erwähnt kann UNEW auch Plots erzeugen. Die möglichen Plots für die primäre Observable sind dabei ein Histogramm und je ein Plot für die integrierte und die normierte Autokorrelationszeit gegen W mit jeweils einer Markierung bei W_{opt} . Handelt es sich nicht mehr um eine primäre Observable, fällt das Histogramm weg. Sollte man Plots erstellen während man Replicas benutzt, wird noch ein zusätzlicher Plot mit einem Histogramm der Replicas erzeugt sowie dessen Wert für Q . Beispiele hierfür finden sich in der Abbildung 1.

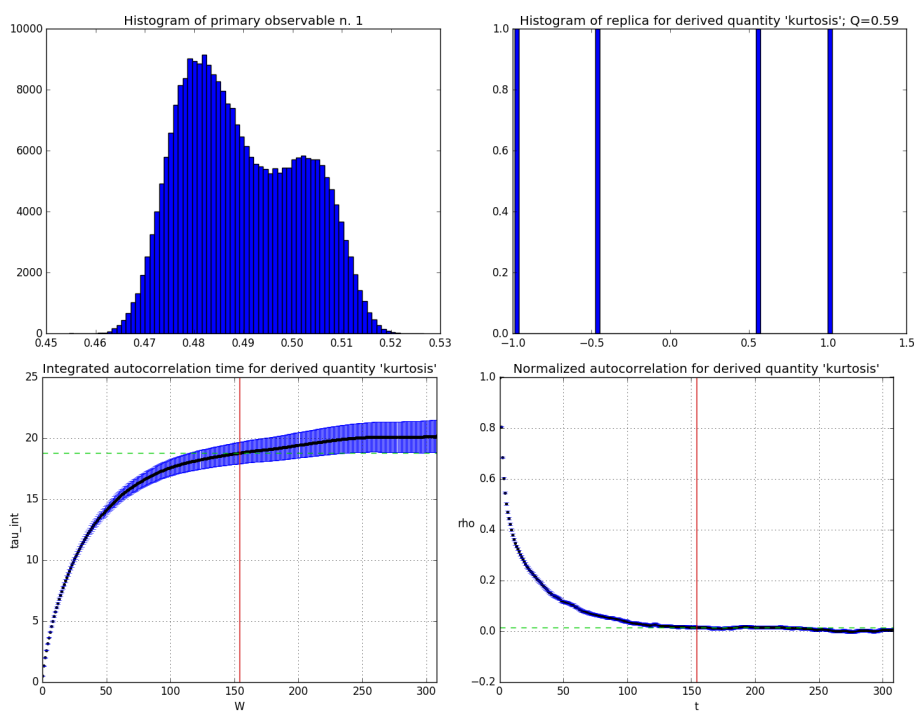


Abbildung 1: Beispiel für die möglichen Plots von UNEW

6 Implementierung

Um das Programm so allgemein wie möglich zu halten, gibt es in dem Verzeichnis von PLASMA ein Template für die lokalen Pfade zu den externen Programmen, die in PLASMA genutzt werden. Dies wurde eingeführt, um die Installation bzw. die Nutzung des Tools auf verschiedenen Computern so einfach wie möglich zu gestalten. Bei einer Neuinstallation werden hier die Pfade zu den externen Programmen angegeben, da diese von Computer zu Computer unterschiedlich sein können. Somit gibt es nur eine Datei, in der man die Pfad eingeben muss, anstatt die entsprechenden Skripte zu suchen, in denen sie eingeführt wurden. Oder noch schlimmer, in jedem Skript einzeln zu ändern. In diesem Template werden außerdem noch die Dateinamen für die erzeugten Dateien mit den Ergebnissen definiert.

Der erste Schritt in der Implementierung bestand also darin, in dieser Template Datei eine neue Variable anzulegen, die den Pfad zu dem UNEW-Tool enthält, welches eingebunden werden soll.

Der nächste Schritt bestand darin, die Analyse des C++ Jackknife-Tools durch UNEW zu ersetzen. Um weiterhin beide Möglichkeiten nutzen zu können, habe ich die Commandlineoption „- analyzeWithUnew“ in arguments.py dem Parser hinzugefügt. Mit dieser ließ sich die Analyse mithilfe von UNEW explizit aktivieren.

Um nun auch UNEW zu nutzen, wurde der Funktion „analyzeDataSet“ eine if-Abfrage hinzugefügt, welche überprüft welche Analysemethode gewünscht ist. Ist UNEW nicht erwünscht, wird die Funktion unverändert ausgeführt.

Sobald aber mithilfe von UNEW gearbeitet werden soll, wird das neue Modul „externalUnew.py“ geladen und die eingelesenen Daten sowie alle Parameter an die Funktion „analyzeDataUsingUnew“ innerhalb des neuen Moduls übergeben. Es wird nun erst einmal geprüft, ob UNEW unter dem angegebenen Verzeichnis in global_defs.py auch verfügbar ist. Wenn dies nicht der Fall ist, wird das Programm mit einer Fehlermeldung beendet.

Ist es verfügbar, wird für die Dauer der Analyse das Modul derivedFuncUnew.py in das angegebene Verzeichnis kopiert, um die Analyse von Suszeptibilität, Skewness und Kurtosis zu ermöglichen.

Die Analyse selbst erfolgt nun in 2 Schritten.

1. Schritt:

Es werden die übergebenen Daten in eine temporäre Datei geschrieben welche, nun eine Spalte mit allen Werten enthält. Es werden auch noch alle nötigen Commandlineoptionen für UNEW ermittelt und in einer Liste gespeichert. UNEW wird nun mithilfe der Funktion subprocess.check_output() als externes Programm gestartet. Der Rückgabewert dieser Funktion ist die Terminalausgabe von UNEW. Aus dieser lassen sich nun die berechneten Werte herausfiltern und in einer Klasse speichern, welche für jeden berechneten Wert ein eigenes Attribut besitzt.

In diesem Schritt wurde nur der Erwartungswert und dessen Fehler berechnet.

2. Schritt:

Als nächstes werden die Commandlineoptionen für die Analyse der höheren Momente sowie die temporäre Dateien, welche nun 3 Spalten $\langle(x - \mu)^n\rangle$ für $n = 2, 3, 4$ enthalten, erstellt

Wieder wird UNEW über subprocess.check_output() gestartet und aus dem

Rückgabewert nun für jeden statistischen Parameter mit allen berechneten Werten eine neue Klasse erzeugt.

Zum Schluss werden wieder alle temporären Dateien gelöscht.

Der Rückgabewert der gesamten Funktion sind allerdings nicht alle Klassen, sondern die alte Klasse `jackknife.obs`.

Dies hat den Grund, dass der Rest des Programms weiterhin das tun soll, was es zuvor auch getan hat. Einige Beispiele wären hier das Erstellen der Plots und Tabellen mit den Ergebnissen. Um dies zu gewährleisten ist es nötig, die alte Klasse weiterhin zu verwenden, um nicht das gesamte PLASMA abzuändern.

Es werden also die erwünschten Werte der Klasse `jackknife.obs` zugewiesen und diese am Ende übergeben. In Listing 3 ist der Codabschnitt für diesen Schritt zu sehen. Zu Beginn wurde die Autokorrelation noch von der ursprünglichen Programmierung übernommen. Im Laufe des Projekts wurden dann mithilfe der sowieso von UNEW berechneten Werte für τ_{int} und der dazugehörige Fehler ebenfalls die Klasse `AutocorrelationInformation` erstellt und auch übergeben.

Probleme gab es vor allem mit der Nutzung der abgeleiteten Funktionen für UNEW. Diese traten auf, da die Funktionen in Modulen gespeichert werden müssen und diese nach dem Starten von UNEW importiert werden müssen. Das Modul mit den Funktionen muss also entweder in den Ordner von UNEW gespeichert werden oder in einem der anderen Verzeichnisse des PYTHONPATH, um es zu importieren. Um diese Funktion zu nutzen, werden also Schreibrechte für das UNEW Verzeichnis benötigt. Dies lässt sich nicht anders lösen ohne den Code von UNEW zu ändern. Das sollte aber vermieden werden, um es anderen Personen zu ermöglichen, den originalen Code zu benutzen.

In dem Modul sind diese Größen nach § 2.4 definiert.

Die Parameter der Funktionen sind jeweils ein Array mit je einem Erwartungswert für jede Spalte der temporären Datei.

Diese Implementierung legt den Grundstein für die anschließende Erweiterungen in PLASMA. Ziel dieser Erweiterungen ist die Möglichkeit, die Analyse mithilfe von Replicas zu ermöglichen.

Hierbei sollen die Verteilungen, die zuvor sowohl einzeln als auch zusammen als eine Verteilung analysiert wurden, nun als unabhängige replicas analysiert werden. Um dies zu ermöglichen, ist ein deutlich tieferer Eingriff in den bisher bestehenden Code von PLASMA nötig. Nach Rücksprache mit Dr. Francesca Cuteri sollte die Analyse mithilfe von Replicas von nun an die Standardmethode von PLASMA werden. Um allerdings die anderen Methoden beizubehalten, wurden die Commandlineoptionen in `arguments.py` erneut verändert. Die erst neu eingeführt Option `analyzeWithUnew` wurde wieder entfernt und zwei neue Optionen hinzugefügt. Diese sind `analyzeWithJackknife` und `analyzeSingleChains`. Es ergeben sich nun also folgende Möglichkeiten in der Verwendung wie in Tabelle 3:

Im ersten Fall bleibt alles wie es ist und es werden alle Verteilungen sowie die zusammengefügte Verteilung mithilfe des Jackknife-Tools analysiert. Im zweiten Fall wird wieder das Jackknife-Tool verwendet, allerdings wird nun nur noch die zusammengefügte Verteilung analysiert. Dies sollte jedoch nach Möglichkeit vermieden werden. In Fall 3 und 4 wird nun die Analyse von UNEW übernommen. In Fall 3 für alle Verteilungen einzeln und in Fall 4 werden die Verteilungen als Replicas verwendet.

Eine wichtige Neuerung besteht in einem geänderten Verhalten des Einlesens

	analyzeWithJackknife	analyzeSingleChains
1	x	x
2	x	-
3	-	x
4	-	-

Tabelle 3: Mögliche Kombinationen für Jackknife und Datenauswahl

der Verteilungen.

Zuvor wird für jeden Wert von β überprüft wie viele Verteilungen existieren sowie für welche eine Analyse ausgeführt werden soll. Im Anschluss wird dann mithilfe einer for-Schleife für jede einzelne Verteilung die Daten eingelesen und die Analyse ausgeführt. Um Replicas nutzen zu können, müssen nun aber die Daten aller Verteilungen eines β Wertes eingelesen werden und nur einmal eine Analyse mit mehreren DataContainern ausgeführt werden. DataContainer sind die Klassen in PLASMA, in denen die Zufallsdaten und alle Parameter gespeichert werden. Die Funktion `readInDataOfSpecificObservableAndBeta(parameters, observable, index, paths)` ist für das Einlesen und Erstellen der Klassen zuständig. Es wird nun eine Liste von Pfaden zu den Verteilungen übergeben und am Schluss wieder eine Liste mit allen erforderlichen DataContainern übergeben. Wieder sollten die alten Analysemethoden weiterhin erhalten bleiben, weshalb wir in diesen Fällen einfach eine Liste mit genau einem DataContainer erhalten.

Wir haben jetzt also eine Liste aller Daten, die wir gleichzeitig analysieren wollen, sei es als Replicas oder wie bisher einzeln mithilfe eines Jackknives oder von UNEW. Für die Programmierung bedeutet das, dass wir alle weiteren Funktionen dahingehend verändern müssen, dass diese Listen von DataContainern akzeptieren.

Dies lässt sich einfach mithilfe von if-Abfragen erreichen, bei denen überprüft wird welche Commandlineoption gewählt wurde. Eine andere Möglichkeit ist die Länge der übergebenen Liste zu überprüfen.

```
def variance(abb):#abb is a list with mean of columns!
    """
    derived function for variance
    <(x-mu)^2>
    """
    return abb[0]

def skewness(abb):
    """
    derived function for skewness
    <(x-mu)^3> / <(x-mu)^2>^(3/2)
    """
    return abb[1]/(abb[0]**(3/2.))

def kurtosis(abb):
    """
    derived function for kurtosis/binder
    <(x-mu)^4> / <(x-mu)^2>^2
```

```
"""  
return abb[2]/(abb[0]**(2))
```

Listing 2: Funktion des kopierten Moduls für UNEW

```
if derived: #only true if at least one derived is  
activated  
    tmpFileDerived=writeTmpFileDerived(dataIn)  
    derivedOutput=subprocess.check_output(["python",  
        local_defs.unew+"/unew-run.py"] + tmpFileDerived +  
        derivedOptions)  
    if(gd.print_out):  
        print(derivedOutput)  
    susc,skew,kur = derivedResults(derivedOutput, args)  
    obs.sus = susc.value  
    obs.sus_err = susc.err  
    obs.skew = skew.value  
    obs.skew_err = skew.err  
    obs.binder = kur.value  
    obs.binder_err = kur.err  
    for _ in tmpFileDerived:  
        os.remove(_)
```

Listing 3: Code für die Analyse der höheren Momente

7 Ergebnisse der Programme

Hier möchte ich nun die Ergebnisse für die verschiedenen Methoden vergleichen. Für diese Zwecke habe ich Zugriff auf echte QCD Daten erhalten, um die Analysen durchzuführen. Die Parameter der mir zur Verfügung stehenden Daten sind in Tabelle 4 zu sehen. Jede der SingleChain-Dateien hat eine Länge von $2 \cdot 10^6$ und die Merged-Dateien entsprechend $8 \cdot 10^6$. Wie schon in § 2 angenommen, sind diese Daten ebenfalls gleichverteilt. Das heißt, wir können alle Daten zur Analyse benutzen und müssen keinen Offset festlegen, welcher die ersten N Werte ignoriert. Bei diesen Daten möchte ich vor allem auf die Obser-

N_f	2
μ_i	0
κ	0.1250
N_τ	8
N_σ	32
β	5.971; 5.982; 5.991

Tabelle 4: Parameter der QCD-Daten

vable „poly_sq“ eingehen. Diese beschreiben die Norm des Polyakov Loop Wert. Hierbei lässt sich der Phasenübergang beobachten. Wir analysieren also die simulierten Daten für alle Temperaturen und suchen nach dem Phasenübergang von Hadronen zum Quark-Gluonen-Plasma. In den Tabellen 9 bis 12 sind die Ergebnisse für die Analyse mithilfe der Jackknifemethode aufgeführt. Die Ergebnisse von UNEW sind in den Tabellen 13 bis 16 für die Analyse der Single Chains zu finden. Die Ergebnisse für die Analyse mithilfe von Replicas sind in Tabelle 17 zu finden. Wie aber findet man einen solchen Phasenübergang und die dazugehörige kritische Temperatur β_c anhand der berechneten Momente m_1, μ_2, B_3 und B_4 ?

Wir betrachten Simulationen, in denen β rund um β_c variiert wird, also von $\beta < \beta_c$ bis hin zu $\beta > \beta_c$. Ist man von dem Phasenübergang noch weit entfernt, hat die Verteilung einen Erwartungswert nahe 0 und ist symmetrisch um diesen. Deshalb ist auch die Skewness $B_3 = 0$. Nähert man sich β_c verformt und verschiebt sich auch die Verteilung, welche wir analysieren, und wird asymmetrisch. Der Erwartungswert erhöht sich also bei dem Phasenübergang auf einen positiven Wert. Nähert man sich von unten an den Phasenübergang an, verschiebt sich die gesamte Verteilung nach rechts und wird dabei rechtsschief, was bedeutet, das die Skewness positiv ist $B_3 > 0$. Nach dem Phasenübergang verschiebt sie sich noch weiter dem Grenzwert entgegen, dabei ist die Skewness jedoch linksschief $B_3 < 0$. Das bedeutet, bei dem kritischen Wert β_c erhalten wir $B_3 = 0$ und die Varianz wird an dieser Stelle maximal. Will man nun die Art des Phasenübergangs bestimmen, benötigt man dazu die Kurtosis B_4 . In diesem Fall ist das allerdings nicht möglich, da wir eine größere Anzahl an Simulationen benötigen, um die Ordnung des Phasenüberganges verlässlich zu bestimmen. Der Grund hierfür liegt in dem räumlich begrenzten Gitter.

7.1 Single Chain Ergebnisse

Im ersten Schritt sehen wir uns die Analyseergebnisse für die einzelnen Ketten, bzw Single Chain Werte, der Temperaturen an und vergleichen die beiden Analyseergebnisse für Jackknife und UNEW.

Schaut man sich die berechneten Erwartungswerte für die verschiedenen Momente der Verteilungen an, sieht man schon mit bloßem Auge die große Ähnlichkeit. Meistens unterscheiden sich diese erst bei der zweiten Nachkommastelle voneinander. Genauer gesagt liegt die maximale Abweichung bei 3%. Wirft man nun einen Blick auf die Fehler der Momente, sieht dies ein wenig anders aus. Hier liegen die Werte aus der UNEW-Analyse bis zu 40% über dem der Jackknife Analyse, wobei auffällt, dass nur ein Fehler der Kurtosis bei $\beta = 5.991$ unter dem Wert der Jackknife-Analyse ist. Die von UNEW berechneten Fehler liegen für Erwartungswert, Varianz und Skewness im Schnitt bei ca. 30% über denen der Jackknife-Methode. Einzig die Fehler für die Kurtosis liegen lediglich ca 12% darüber. Betrachten wir nun die Ergebnisse mithilfe des Gauß-Test aus Abschnitt 2.5, lässt sich die große Ähnlichkeit der Schätzwerte bestätigen. Hierfür wurden jeweils die Ergebnisse jeder Kette für beide Methoden verglichen. Keiner der Werte liegt unter 87% und ein Großteil sogar über 99%. Zu sehen sind die Werte in Tabelle 5. Wobei nach S.60 in [1] ein Wert für Q sehr nahe an 1 ein Hinweis dafür ist, dass die Fehler überschätzt wurden. Hier kann man jedoch davon ausgehen, dass die Fehler des Jackknifes eher unterschätzt wurden und somit die Fehler von UNEW für den Gauß-Test als überschätzt wahrgenommen wurden. Betrachten wir nun die von PLASMA erstellen „multi-

#beta	seed	Mean	Variance	Skew	Kurtosis
5.971	821	99.99%	100.00%	99.98%	99.96%
5.971	1932	99.58%	99.53%	98.10%	99.35%
5.971	2043	97.05%	99.98%	99.64%	99.99%
5.971	9710	99.16%	99.61%	99.99%	99.21%
5.981	921	99.50%	99.59%	99.98%	99.25%
5.981	1032	99.98%	99.24%	99.97%	98.88%
5.981	2143	97.06%	99.96%	99.15%	99.99%
5.981	9810	98.30%	99.98%	99.78%	99.98%
5.991	21	98.79%	99.96%	99.89%	99.92%
5.991	1132	95.32%	93.87%	95.22%	87.86%
5.991	2243	99.91%	99.93%	99.69%	99.59%
5.991	9910	99.22%	99.45%	99.39%	99.96%

Tabelle 5: Ergebnis des Gauß-Tests für SingleChain Jackknife und UNEW

pleDatasets“-Plots. Es existiert jeweils ein Plot pro berechnetem Schätzwert. In ihm sind die berechneten Werte und deren Fehler dieses Moments für jede Kette gegen β aufgetragen. Die zusätzlich angegebenen Werte n_σ haben eine ähnliche Aussagekraft wie Q des Gauß-Tests. n_σ entspricht nämlich d' in Gl. (43). Geht man von $Q_{cut} = 0.05$ aus, entspricht dies für $n_{\sigma,cut} \approx 1.95996$. Werte $n_\sigma > n_{\sigma,cut}$ sind äquivalent für $Q < Q_{cut}$. Für die Plots werden alle Ergebnisse eines β 's untereinander mit dieser Methode verglichen und der größte Wert ausgewählt. (Dies geschieht in der Funktion `findMaxSigmaDiscrepancy(dataOfBeta, errorOfBeta)` innerhalb des Moduls `plotting.py`)

Natürlich werden in einem Plot nur die Ergebnisse innerhalb der Γ - oder der Jackknife-Methode miteinander verglichen, jedoch lassen sich dann die Plots gut vergleichen. Die einzelnen Ketten wurden mit den gleichen Parametern erzeugt, weshalb auch hier der Gauß-Test verwendet werden kann. Dieser liefert dann Aufschluss darüber, wie nahe die Schätzwerte der jeweiligen Ketten sich untereinander sind. Die Plots sind in den Abbildungen 2 für die Jackknife Methode und 3 für UNEW zu sehen für die Skewness zu sehen. Es wurden hier die Plots der Skewness stellvertretend für alle Momente dargestellt, da die Ergebnisse alle sehr ähnlich sind.

Wie man gut sehen kann, liegen die n_σ Werte von UNEW teilweise sehr deutlich unter denen des Jackknifes. Es liegen generell alle n_σ -Werte aller Momente des Jackknifes über jenen von UNEW.

Wenden wir uns nun der berechneten Autokorrelationszeit von PLASMA und UNEW zu. Hierbei fallen zwei Dinge besonders auf: Das Erste, was einem auffällt sind die Autokorrelationswerte für die Kurtosis. Hier liegen die von PLASMA deutlich mit fast einer Größenordnung über denen von UNEW. Das gleiche gilt auch für deren Fehler.

Grund hierfür ist die unterschiedliche Implementierung bei der Bestimmung von τ_{int} . Während UNEW diese direkt über die gegebene Verteilungen berechnet, nimmt PLASMA den maximalen Wert aus beiden zentralen Momenten. In diesem Fall ist τ_{int} des zweiten zentralen Moments größer als τ_{int} des vierten zentralen Moments. Deshalb stimmen die Autokorrelationszeiten der Kurtosis des Jackknifes auch mit denen der Varianz überein. (Siehe Tabellen 10 und 12) Das Zweite fällt in allen anderen Fällen auf: Dort sind die Werte für τ_{int} im Durchschnitt um die 10% höher als jene von PLASMA.

Aber bei welchem Wert für β liegt nun der Phasenübergang?

Wie wir wissen müssen wir hierfür die Stelle des Vorzeichenwechsels der Skewness finden. Hierfür können wir wieder unsere MultipleDatasets-Plots in Abbildung 2 und 3 verwenden. Wie man nun sehen kann, fällt die Skewness von $\beta = 5.971$ bis $\beta = 5.991$ deutlich ab. Für $\beta = 5.981$ sind die Werte der einzelnen Ketten rund um die 0 verteilt, wobei 3 von 4 Werten unterhalb und nur einer über der 0 liegt. Damit können wir nun schon sagen, dass der Phasenübergang bei der kritischen Temperatur $\beta_c = 5.981$ stattgefunden hat.

Weiterhin lassen sich nun auch die anderen Hinweise auf einen Phasenübergang überprüfen. Zum einen soll hierfür der Erwartungswert deutlich ansteigen. Schaut man sich nun die Werte an, fällt auf, dass sich diese zwischen $\beta = 5.971$ und $\beta = 5.991$ mehr als verdoppeln. Auch dies ist ein Hinweis auf einen Phasenübergang.

Zum anderen soll die Varianz während eines Phasenüberganges ein Maximum annehmen. Wirft man nun wieder einen Blick auf die Werte, kann man auch bei $\beta = 5.981$ sehen, dass die Werte über denen der anderen β 's liegen. Allerdings fällt auf, dass die Werte für $\beta = 5.991$ extrem weit streuen. Für die Jackknife-Methode liegen sie sogar über dem geforderten Wert für $n_{\sigma, cut}$. Nichtsdestotrotz ist dies wieder ein Hinweis auf die gleiche Stelle für den Phasenübergang.

Wir können also sagen, dass der Phasenübergang bei oder sehr nahe um $\beta = 5.981$ stattgefunden hat. Mit einer höheren Auflösung von β ließe es sich evtl. noch genauer feststellen, jedoch mit deutlich höherem Rechenaufwand der Simulationen.

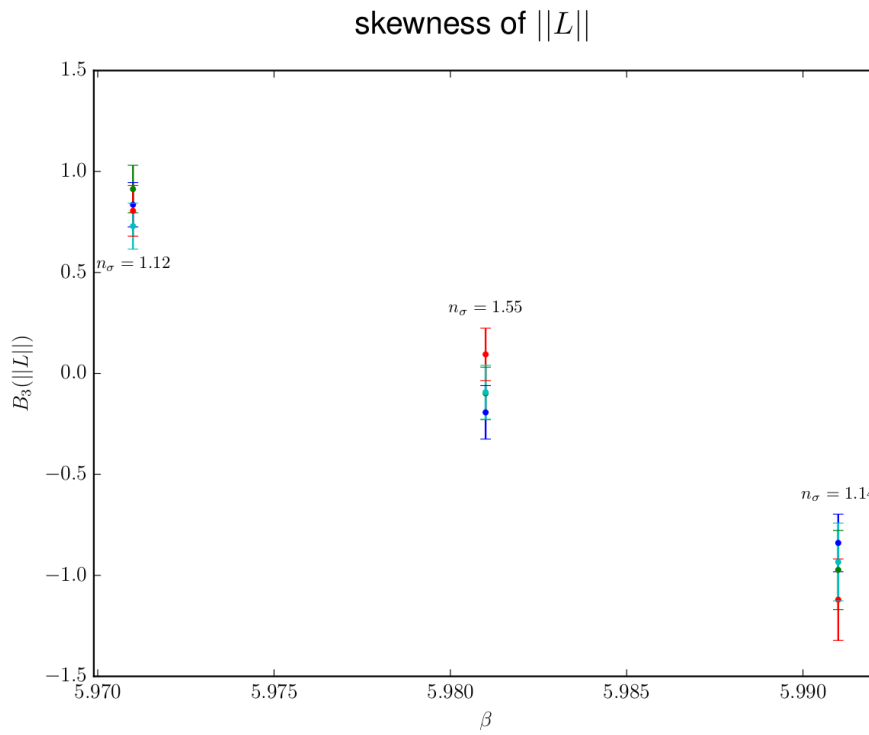


Abbildung 2: MultipleDatasets für Skewness mithilfe des Jackknifes

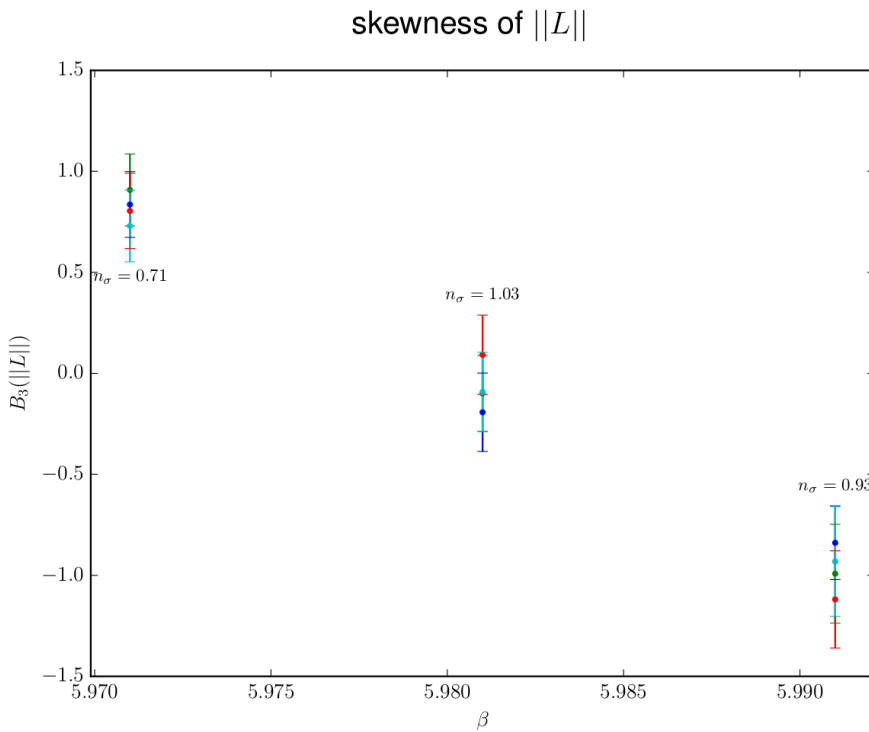


Abbildung 3: MultipleDatasets für Skewness mithilfe UNEW's

7.2 Merged Chain vs Replica

Wir vergleichen nun die Ergebnisse der Mergedchains des Jackknife mit den Ergebnissen der Replicaanalyse von UNEW. Für die Merged Dateien werden alle SingleChain Dateien hintereinander in eine Datei geschrieben und als eine einzige Verteilung analysiert. Mit UNEW lassen sich all diese Verteilungen gleichzeitig als sogenannte Replicas analysieren.

Die Daten dazu sind für UNEW in Tabelle 17 und für's Jackknife in Tabelle 18 zu finden.

Auch hier unterscheiden sich die Schätzwerte aller Momente, bis auf eine Ausnahme, erst bei der 2. Nachkommastelle. Die prozentuale Abweichung liegt dabei unter 1%. Die Ausnahme bilden hierbei die Werte für die Skewness. Hier liegt die Abweichung eines Wertes einmal bei 8%. Die Fehler der Schätzwerte verhalten sich ähnlich zu denen der Single Chain Analyse. Auch hier sind wieder alle von UNEW berechneten Fehler um ca 30% größer als die Fehler des Jackknifes. Wobei auch hier wieder die Fehler der Kurtosis mit ca. 14% geringer ausfallen als der Rest. Berechnet man nun auch für diese Analyse die Q -Werte nach der Gauß-Methode, erhält man die Ergebnisse aus Tabelle 6. Man erkennt nun, dass die Q -Werte nicht mehr alle so nahe an der 1 sind, wie es noch mithilfe der SingleChain Analyse der Fall war. Ausnahme bietet hierbei die Spalte, in der die Werte für den Erwartungswert verglichen wurden. Hier sind alle Werte genau 1. Wie schon im Abschnitt zuvor, spricht dies für eine Überschätzung der Fehler bei den Werten bzw. wurden die Fehler der Jackknife-Methode unterschätzt. Dass es nun keine Plots mehr für n_σ -Wertegibt, sollte klar sein, denn es existiert ja jetzt pro β nur noch ein Ergebnis.

Auch die Ergebnisse der Autokorrelationszeiten ähneln den Ergebnissen der SingleChain Analyse. Für Erwartungswert und Varianz sind auch hier die Werte von UNEW um ca 10% über denen von PLASMA. Für Skewness und Kurtosis gilt dies nicht, da hier τ_{int} von PLASMA über denen von UNEW ist. Der Grund hierfür ist der gleiche wie bei den SingleChains, es liegt an der Implementierung. Bei der Skewness liegt die prozentuale Abweichung bis zu 30% darunter und bei der Kurtosis sogar bei fast 90%. Auch hier können wir wieder die Ähnlichkeiten zum vorherigen Abschnitt feststellen.

Untersuchen wir nun auch diese Analyse auf den Phasenübergang. Der Vorzeichenwechsel der Skewness ist wieder deutlich bei beiden Ergebnissen zu erkennen. Die beiden Werte der äußeren β 's liegen in derselben Größenordnung von der 0 entfernt und der mittlere Wert bei $\beta = 5.981$ knapp unterhalb der 0. Ebenso kann man wieder gut die Verdoppelung des Erwartungswertes erkennen sowie das Maximum der Varianz bei $\beta = 5.981$. Wir können also wieder davon ausgehen, dass der Phasenübergang bei $\beta_c = 5.981$ stattgefunden hat.

#beta	Mean	Variance	Skew	Kurtosis
5.971	100.00%	97.45%	98.55%	90.02%
5.981	100.00%	84.81%	95.94%	59.31%
5.991	100.00%	94.87%	93.53%	85.23%

Tabelle 6: Ergebniss des Gauß-Tests für mergedchains in Jackknife und UNEW mit Replicas

7.3 Andere Observablen

Die Beobachtungen, die die bei der Observablen „poly_sq“ gemacht wurden, finden sich in einer ähnlichen Form auch bei allen anderen poly-Observablen. Natürlich stimmen nicht immer alle Abweichungen exakt überein, da poly_sq ja sowohl die imaginären als auch die reellen Werte der Polyakov Loop beachtet. Natürlich lassen sich diese Werte auch allein analysieren. Die berechneten Pla_q-Werte weichen von den Beobachtungen ein wenig ab. Hierbei stimmen die Werte der Momente noch deutlich überein. Häufig existiert sogar gar keine Abweichung zwischen den beiden Methoden.

Einen Unterschied gibt es jedoch bei den angegebenen Fehlern der Momente: Hier sind alle von UNEW angegebenen Fehler bis zu doppelt so groß wie jene von PLASMA.

7.4 Übereinstimmung schlägt fehl

Analysiert man die Daten mit den Parametern aus Tabelle 7 mithilfe von Replicas bzw. Merged Dateien mithilfe des Jackknives, erhält man die Ergebnisse in den Tabelle 19 und 20 Benutzt man nun diese Ergebnisse um auch hier wieder den Gauß-Test durchzuführen erhält man die Ergebnisse in Tabelle 8. Betracht-

nflavors	2
mui	0
kappa	1000
nt	8
ns	24
beta	6.035; 6.040

Tabelle 7: Parameter der zu kurzen QCD-Daten

#beta	Mean	Variance	Skewness	Kurtosis
6.035	100.00%	72.37%	87.19%	78.41%
6.04	100.00%	38.84%	90.20%	2.85%

Tabelle 8: Ergebniss des Gauß-Tests für mergedchains in Jackknife und UNEW mit Replicas mit Fehlschlag

tet man nun diese Werte fällt natürlich direkt auf, dass einer der Werte für die Kurtosis unterhalb von 5% liegt. Deshalb stellt sich auch direkt die Frage woran die Abweichung der beiden Werte liegt. Hierfür gibt es drei Möglichkeiten. Entweder einer der beiden Analysemethoden hat einen anderen Wert berechnet oder Grund für diese Abweichung liegt in den Daten. Überprüfen wir zunächst einmal die MultipleDataset-Plots die für eine SingleChain Analyse erstellt werden und vergleichen diese für die beiden Methoden miteinander in den Abbildungen 4 und 5. Wie wir nun sehen liegen die Ergebnisse für die einzelnen Ketten vor allem bei $\beta = 6.04$ sehr weit voneinander entfernt. Dies bestätigt auch den

angegebene Wert von n_σ bei beiden Methoden. Dieser liegt in beiden Fällen weit über dem geforderten Wert für $n_{\sigma,cut}$. Bei den Merged Ergebnissen des Jackknifes fügt man aber alle Ketten zusammen unter der Annahme, dass diese die gleiche Verteilung beschreiben. Wir wissen nun aber, dass diese Ketten stark voneinander Abweichen. UNEW ist also in diesem Fall zuverlässiger mithilfe der Replicaanalyse. Die Verteilungen ließen sich verbessern, indem man ihre Länge weiter erhöht. Dadurch würden sich die Werte der einzelnen Ketten weiter annähern und auch wieder konsistente Werte erhalten.

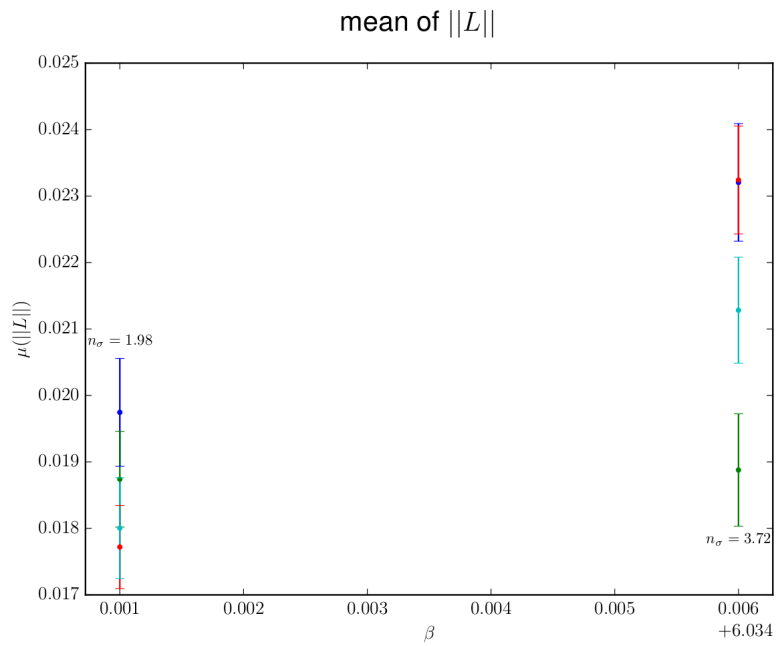


Abbildung 4: MultipleDatasets für Mean mithilfe des Jackknifes

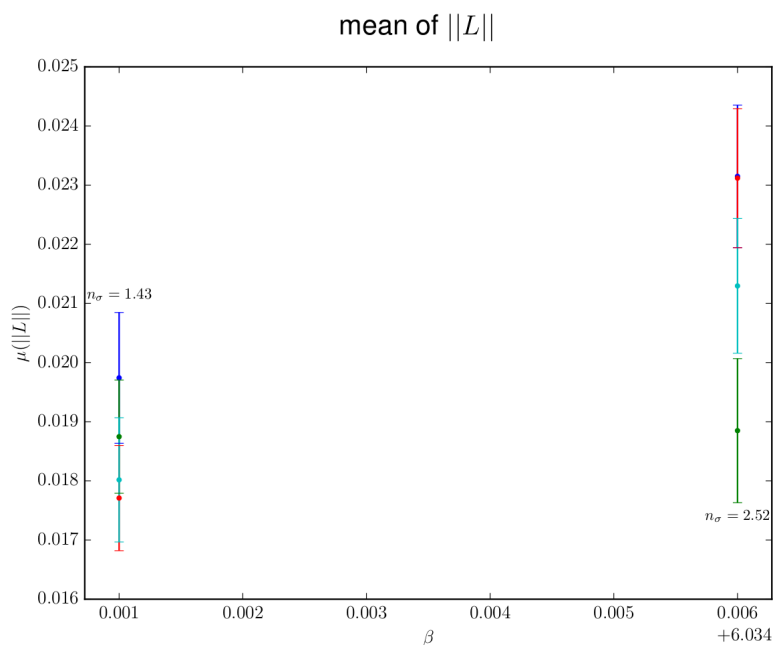


Abbildung 5: MultipleDatasets für Mean mithilfe UNEW's

8 Fazit

In dieser Arbeit wurde PLASMA verändert, um die Analyse der Observablen durch ein weiteres externes Open Source Programm durchführen zu können. Dabei wurde das Programm nur erweitert, also alle bisher bekannten Methoden und externen Programme sind weiterhin verfügbar. Bei der Benutzung wird nun standardmäßig die Analyse mithilfe von UNEW unter der Verwendung von Replicas benutzt. Es fällt bei der Analyse der Ergebnisse auf, dass die Ergebnisse für die Autokorrelationszeiten der Kurtosis aufgrund der unterschiedlichen Implementierungen bzw. Benutzungen von τ_{int} , nachdem dieses mithilfe der Γ -Methode berechnet wurde, sehr deutlich voneinander abweichen.

Auffällig ist auch, dass die Fehler mithilfe von UNEW oft über denen von PLASMA liegen. Die Γ -Methode strebt jedoch an, die Fehler der Fehler zu verringern, worauf schon der Titel von Ulli Wolff's Veröffentlichung „Monte Carlo errors with less errors“ [4] hinweist. Nach dem Vergleich der verschiedenen Analysemethoden lässt sich gut sagen, dass sich die Ergebnisse sehr ähnlich sind und die Phasenübergänge an den gleichen Stellen identifiziert werden konnten.

PLASMA ließe sich noch weiter verbessern, indem man die Berechnungen auf verschiedenen Prozessorkerne aufspalten würde. Die Zeit für die Analyse würde sich dadurch erheblich verbessern. Die Tatsache, dass hierfür unterschiedliche Daten analysiert werden, ist für diese Art der Programmierung sehr geeignet. Allerdings sind dafür erhebliche Änderungen an PLASMA nötig, da häufig die Ergebnisse schon zwischendurch in die Ausgabedaten geschrieben werden. Generell merkt man, sobald man sich mit dem Code von PLASMA auseinandersetzt, dass das Programm immer wieder erweitert worden ist und einige Klassen oft für Dinge benutzt werden, für die sie anfangs nicht geplant waren. Hierdurch entstehen oft Probleme, die durch workarounds umgangen werden. Ein Beispiel hierfür ist meiner Meinung nach die Klasse dataContainer und die verschiedenen globalArray Variablen.

Die Arbeit, diese umzuschreiben geht jedoch über diese Bachelorarbeit hinaus, jedoch würden sie die Änderungen für MultiCore Berechnungen sehr erleichtern und den Programmcode übersichtlicher gestalten.

Abschließend möchte ich mich noch bei Prof. Dr. Owe Philipsen, für die Möglichkeit diese Bachelorarbeit bei ihm schreiben zu dürfen, bedanken. Mein Dank gilt ebenso Dr. Francesca Cuteri, die mir in dieser Zeit bei Fragen oder Problemen immer sehr hilfreich zu Seite stand und mich damit sehr unterstützt hat.

A Anhang

#beta	seed	mean	+ -	tau	+ -	binsize
5.971	merged	1.08E-02	2.68E-04	1.09E+03	1.39E+02	2176
5.971	821	1.06E-02	4.62E-04	8.07E+02	1.58E+02	1626
5.971	1932	1.05E-02	5.56E-04	1.07E+03	2.33E+02	2148
5.971	2043	1.08E-02	4.55E-04	8.29E+02	1.64E+02	1658
5.971	9710	1.14E-02	5.88E-04	1.18E+03	2.66E+02	2370
5.981	merged	1.82E-02	3.69E-04	1.12E+03	1.45E+02	2266
5.981	921	1.89E-02	7.08E-04	1.07E+03	2.32E+02	2140
5.981	1032	1.85E-02	7.00E-04	1.02E+03	2.17E+02	2040
5.981	2143	1.70E-02	8.32E-04	1.36E+03	3.21E+02	2714
5.981	9810	1.83E-02	6.99E-04	1.11E+03	2.43E+02	2214
5.991	merged	2.62E-02	2.87E-04	9.42E+02	1.13E+02	1900
5.991	21	2.54E-02	5.85E-04	7.58E+02	1.45E+02	1518
5.991	1132	2.60E-02	6.15E-04	1.07E+03	2.32E+02	2138
5.991	2243	2.70E-02	5.68E-04	1.02E+03	2.17E+02	2040
5.991	9910	2.64E-02	4.68E-04	8.09E+02	1.58E+02	1620

Tabelle 9: Ergebnisse für Mean mithilfe der Jackknifemethode

#beta	seed	variance	+ -	tau	+ -	binsize
5.971	merged	5.04E-05	1.67E-06	5.10E+02	4.77E+01	1020
5.971	821	4.86E-05	3.19E-06	4.31E+02	6.61E+01	1000
5.971	1932	5.08E-05	3.86E-06	6.40E+02	1.14E+02	1280
5.971	2043	4.79E-05	3.08E-06	4.30E+02	6.60E+01	862
5.971	9710	5.38E-05	3.07E-06	4.01E+02	5.99E+01	806
5.981	merged	8.27E-05	1.42E-06	2.58E+02	1.80E+01	516
5.981	921	8.38E-05	3.09E-06	2.94E+02	3.87E+01	604
5.981	1032	7.88E-05	2.93E-06	2.53E+02	3.13E+01	506
5.981	2143	8.61E-05	2.71E-06	2.28E+02	2.71E+01	464
5.981	9810	8.00E-05	2.86E-06	2.70E+02	3.44E+01	542
5.991	merged	6.29E-05	3.65E-06	9.21E+02	1.10E+02	1842
5.991	21	7.21E-05	6.67E-06	6.80E+02	1.25E+02	1408
5.991	1132	6.48E-05	8.30E-06	1.17E+03	2.61E+02	2336
5.991	2243	6.02E-05	8.49E-06	1.07E+03	2.33E+02	2150
5.991	9910	5.19E-05	5.55E-06	6.39E+02	1.14E+02	1278

Tabelle 10: Ergebnisse für Varianz mithilfe der Jackknifemethode

#beta	seed	skewness	+-	tau	+-	binsize
5.971	merged	8.20E-01	5.92E-02	6.30E+02	6.42E+01	1260
5.971	821	8.36E-01	1.10E-01	4.98E+02	8.09E+01	1000
5.971	1932	9.13E-01	1.18E-01	7.30E+02	1.37E+02	1462
5.971	2043	8.05E-01	1.26E-01	5.45E+02	9.17E+01	1092
5.971	9710	7.30E-01	1.14E-01	5.30E+02	8.82E+01	1062
5.981	merged	-7.53E-02	6.59E-02	7.75E+02	8.61E+01	1584
5.981	921	-1.93E-01	1.32E-01	7.80E+02	1.51E+02	1562
5.981	1032	-9.87E-02	1.30E-01	6.75E+02	1.23E+02	1350
5.981	2143	9.45E-02	1.30E-01	7.68E+02	1.47E+02	1536
5.981	9810	-9.29E-02	1.33E-01	7.33E+02	1.38E+02	1468
5.991	merged	-9.80E-01	9.03E-02	9.88E+02	1.21E+02	1978
5.991	21	-8.39E-01	1.43E-01	7.24E+02	1.36E+02	1470
5.991	1132	-9.73E-01	1.96E-01	1.29E+03	2.99E+02	2580
5.991	2243	-1.12E+00	2.02E-01	1.11E+03	2.45E+02	2226
5.991	9910	-9.34E-01	1.93E-01	7.44E+02	1.41E+02	1488

Tabelle 11: Ergebnisse für Skewness mithilfe der Jackknifemethode

#beta	seed	kurtosis	+-	tau	+-	binsize
5.971	merged	2.88E+00	5.43E-02	5.10E+02	4.77E+01	1020
5.971	821	2.93E+00	1.01E-01	4.31E+02	6.61E+01	1000
5.971	1932	3.02E+00	1.06E-01	6.40E+02	1.14E+02	1280
5.971	2043	2.93E+00	1.43E-01	4.30E+02	6.60E+01	862
5.971	9710	2.68E+00	8.68E-02	4.01E+02	5.99E+01	806
5.981	merged	1.92E+00	2.17E-02	2.58E+02	1.80E+01	516
5.981	921	1.95E+00	4.46E-02	2.94E+02	3.87E+01	604
5.981	1032	2.00E+00	4.69E-02	2.53E+02	3.13E+01	506
5.981	2143	1.84E+00	4.02E-02	2.28E+02	2.71E+01	464
5.981	9810	1.98E+00	4.50E-02	2.70E+02	3.44E+01	542
5.991	merged	3.69E+00	7.32E-02	9.21E+02	1.10E+02	1842
5.991	21	3.15E+00	1.13E-01	6.80E+02	1.25E+02	1408
5.991	1132	3.60E+00	1.35E-01	1.17E+03	2.61E+02	2336
5.991	2243	4.11E+00	1.78E-01	1.07E+03	2.33E+02	2150
5.991	9910	4.01E+00	2.36E-01	6.39E+02	1.14E+02	1278

Tabelle 12: Ergebnisse für Kurtosis mithilfe der Jackknifemethode

#beta	seed	mean	+-	tau	+-	binsize
5.971	merged	1.08E-02	3.95E-04	1.23E+03	2.05E+02	2476
5.971	821	1.06E-02	7.63E-04	1.18E+03	3.27E+02	2370
5.971	1932	1.05E-02	8.02E-04	1.25E+03	3.52E+02	2500
5.971	2043	1.08E-02	6.59E-04	8.98E+02	2.25E+02	1798
5.971	9710	1.14E-02	8.38E-04	1.29E+03	3.66E+02	2580
5.981	merged	1.82E-02	4.82E-04	1.12E+03	1.79E+02	2244
5.981	921	1.89E-02	9.80E-04	1.13E+03	3.08E+02	2270
5.981	1032	1.85E-02	8.77E-04	9.68E+02	2.49E+02	1938
5.981	2143	1.70E-02	1.11E-03	1.41E+03	4.14E+02	2822
5.981	9810	1.83E-02	9.39E-04	1.09E+03	2.92E+02	2182
5.991	merged	2.62E-02	3.93E-04	9.81E+02	1.49E+02	1970
5.991	21	2.54E-02	7.24E-04	7.21E+02	1.66E+02	1442
5.991	1132	2.59E-02	9.62E-04	1.38E+03	4.03E+02	2766
5.991	2243	2.70E-02	7.92E-04	1.03E+03	2.71E+02	2066
5.991	9910	2.65E-02	7.04E-04	9.45E+02	2.41E+02	1892

Tabelle 13: Ergebnisse für Mean mithilfe von UNEW

#beta	seed	variance	+-	tau	+-	binsize
5.971	merged	5.04E-05	2.65E-06	5.85E+02	7.17E+01	1172
5.971	821	4.86E-05	4.81E-06	5.03E+02	1.01E+02	1008
5.971	1932	5.08E-05	6.50E-06	8.04E+02	1.93E+02	1608
5.971	2043	4.79E-05	4.57E-06	4.67E+02	9.14E+01	952
5.971	9710	5.38E-05	4.71E-06	4.55E+02	8.80E+01	930
5.981	merged	8.27E-05	2.11E-06	2.83E+02	2.55E+01	566
5.981	921	8.38E-05	5.23E-06	4.08E+02	7.56E+01	816
5.981	1032	7.88E-05	4.10E-06	2.69E+02	4.24E+01	540
5.981	2143	8.61E-05	3.81E-06	2.32E+02	3.44E+01	464
5.981	9810	8.00E-05	4.49E-06	3.21E+02	5.42E+01	660
5.991	merged	6.29E-05	5.05E-06	9.60E+02	1.44E+02	1932
5.991	21	7.21E-05	8.85E-06	6.98E+02	1.59E+02	1396
5.991	1132	6.60E-05	1.31E-05	1.52E+03	4.56E+02	3036
5.991	2243	6.02E-05	1.08E-05	1.03E+03	2.71E+02	2064
5.991	9910	5.19E-05	7.82E-06	7.49E+02	1.75E+02	1500

Tabelle 14: Ergebnisse für Varianz mithilfe von UNEW

#beta	seed	skewness	+-	tau	+-	binsize
5.971	merged	8.20E-01	9.17E-02	6.00E+02	7.43E+01	1202
5.971	821	8.36E-01	1.62E-01	4.55E+02	8.80E+01	930
5.971	1932	9.08E-01	1.78E-01	6.25E+02	1.37E+02	1250
5.971	2043	8.04E-01	1.87E-01	5.44E+02	1.13E+02	1090
5.971	9710	7.30E-01	1.77E-01	5.84E+02	1.25E+02	1176
5.981	merged	-7.53E-02	9.42E-02	7.43E+02	1.01E+02	1498
5.981	921	-1.93E-01	1.94E-01	7.94E+02	1.90E+02	1600
5.981	1032	-9.88E-02	1.88E-01	6.79E+02	1.53E+02	1358
5.981	2143	9.20E-02	1.96E-01	8.90E+02	2.22E+02	1782
5.981	9810	-9.22E-02	1.97E-01	7.58E+02	1.78E+02	1526
5.991	merged	-9.80E-01	1.15E-01	6.79E+02	8.84E+01	1358
5.991	21	-8.39E-01	1.82E-01	5.12E+02	1.04E+02	1024
5.991	1132	-9.92E-01	2.45E-01	8.88E+02	2.21E+02	1778
5.991	2243	-1.12E+00	2.41E-01	7.21E+02	1.66E+02	1442
5.991	9910	-9.31E-01	2.71E-01	6.11E+02	1.32E+02	1222

Tabelle 15: Ergebnisse für Skewness mithilfe von UNEW

#beta	seed	kurtosis	+-	tau	+-	binsize
5.971	merged	2.89E+00	5.81E-02	8.21E+01	4.30E+00	200
5.971	821	2.93E+00	1.13E-01	6.66E+01	5.90E+00	156
5.971	1932	3.02E+00	1.13E-01	8.02E+01	7.60E+00	160
5.971	2043	2.93E+00	1.58E-01	1.18E+02	1.32E+01	236
5.971	9710	2.68E+00	9.53E-02	7.72E+01	7.20E+00	156
5.981	merged	1.92E+00	3.04E-02	1.85E+02	1.39E+01	374
5.981	921	1.95E+00	5.83E-02	1.72E+02	2.25E+01	344
5.981	1032	2.00E+00	6.56E-02	1.90E+02	2.60E+01	386
5.981	2143	1.84E+00	5.40E-02	1.56E+02	1.97E+01	320
5.981	9810	1.98E+00	6.47E-02	1.84E+02	2.48E+01	378
5.991	merged	3.69E+00	7.60E-02	1.16E+02	7.20E+00	246
5.991	21	3.15E+00	1.20E-01	1.26E+02	1.45E+01	250
5.991	1132	3.57E+00	1.39E-01	1.20E+02	1.35E+01	246
5.991	2243	4.11E+00	1.84E-01	1.25E+02	1.44E+01	250
5.991	9910	4.01E+00	2.04E-01	1.16E+02	1.29E+01	246

Tabelle 16: Ergebnisse für Kurtosis mithilfe von UNEW

#beta	mean	+-	tau	+-	binsize
mean					
5.971	1.08E-02	3.94E-04	1.23E+03	2.03E+02	2460
5.981	1.82E-02	4.84E-04	1.13E+03	1.81E+02	2260
5.991	2.62E-02	3.95E-04	9.89E+02	1.50E+02	1980
variance					
5.971	5.03E-05	2.65E-06	5.86E+02	7.18E+01	1170
5.981	8.22E-05	2.19E-06	3.01E+02	2.80E+01	604
5.991	6.25E-05	5.04E-06	9.68E+02	1.46E+02	1940
skewness					
5.971	8.18E-01	9.28E-02	6.09E+02	7.59E+01	1230
5.981	-6.94E-02	9.55E-02	7.46E+02	1.01E+02	1490
5.991	-9.68E-01	1.17E-01	6.85E+02	8.96E+01	1370
kurtosis					
5.971	2.89E+00	5.84E-02	8.24E+01	4.40E+00	168
5.981	1.94E+00	3.05E-02	1.79E+02	1.33E+01	358
5.991	3.67E+00	7.86E-02	1.23E+02	7.80E+00	250

Tabelle 17: Ergebnisse mit Replicas

#beta	mean	+-	tau	+-	binsize
mean					
5.971	1.08E-02	2.68E-04	1.09E+03	1.39E+02	2180
5.981	1.82E-02	3.69E-04	1.12E+03	1.45E+02	2270
5.991	2.62E-02	2.87E-04	9.42E+02	1.13E+02	1900
variance					
5.971	5.04E-05	1.67E-06	5.10E+02	4.77E+01	1020
5.981	8.27E-05	1.42E-06	2.58E+02	1.80E+01	516
5.991	6.29E-05	3.65E-06	9.21E+02	1.10E+02	1840
skewness					
5.971	8.20E-01	5.92E-02	6.30E+02	6.42E+01	1260
5.981	-7.53E-02	6.59E-02	7.75E+02	8.61E+01	1580
5.991	-9.80E-01	9.03E-02	9.88E+02	1.21E+02	1980
kurtosis					
5.971	2.88E+00	5.43E-02	5.10E+02	4.77E+01	1020
5.981	1.92E+00	2.17E-02	2.58E+02	1.80E+01	516
5.991	3.69E+00	7.32E-02	9.21E+02	1.10E+02	1840

Tabelle 18: Ergebnisse der MergedChains mithilfe des Jackknifes

#beta	value	+-	tau	+-
mean				
6.04E+00	1.86E-02	3.75E-04	4.79E+02	5.81E+01
6.04E+00	2.16E-02	4.35E-04	6.30E+02	8.53E+01
variance				
6.04E+00	1.04E-04	1.59E-06	9.11E+01	5.50E+00
6.04E+00	1.15E-04	1.76E-06	1.03E+02	6.60E+00
skewness				
6.04E+00	2.31E-01	5.67E-02	2.64E+02	2.50E+01
6.04E+00	-5.32E-02	6.30E-02	3.63E+02	3.93E+01
kurtosis				
6.04E+00	2.04E+00	2.19E-02	9.11E+01	5.50E+00
6.04E+00	1.96E+00	1.98E-02	1.03E+02	6.60E+00

Tabelle 19: Ergebnisse der MergedChains mithilfe des Jackknifes des alternativen Datensatzes

#beta	value	+-	tau	+-
mean				
6.04E+00	1.86E-02	5.05E-04	4.89E+02	7.40E+01
6.04E+00	2.16E-02	6.85E-04	8.12E+02	1.50E+02
variance				
6.04E+00	1.03E-04	2.34E-06	9.68E+01	7.40E+00
6.04E+00	1.12E-04	3.00E-06	1.38E+02	1.23E+01
skewness				
6.04E+00	2.14E-01	8.89E-02	2.97E+02	3.66E+01
6.04E+00	-3.78E-02	1.08E-01	4.22E+02	6.01E+01
kurtosis				
6.04E+00	2.05E+00	2.92E-02	4.38E+01	2.40E+00
6.04E+00	2.04E+00	3.07E-02	6.94E+01	4.60E+00

Tabelle 20: Ergebnisse mithilfe von Replicas des alternativen Datensatzes

Literatur

- [1] B. Berg. *Markov Chain Monte Carlo Simulations and Their Statistical Analysis: With Web-based Fortran Code*. World Scientific, 2004.
- [2] B. De Palma, M. Erba, L. Mantovani, and N. Mosco. A Python program for the implementation of the Γ -method for Monte Carlo simulations. 2017.
- [3] A. Sciarra. *The QCD phase diagram at purely imaginary chemical potential from the lattice*. PhD thesis, 2016.
- [4] U. Wolff. Monte Carlo errors with less errors. *Comput. Phys. Commun.*, 156:143–153, 2004. [Erratum: *Comput. Phys. Commun.*176,383(2007)].

Abbildungsverzeichnis

Deckblatt: Logo Goethe Uni entnommen von <https://de.wikipedia.org/wiki/Datei:Goethe-Logo.svg>

1	Beispiel für die möglichen Plots von UNEW	22
2	MultipleDatasets für Skewness mithilfe des Jackknifes	30
3	MultipleDatasets für Skewness mithilfe UNEW's	30
4	MultipleDatasets für Mean mithilfe des Jackknifes	34
5	MultipleDatasets für Mean mithilfe UNEW's	34

Tabellenverzeichnis

1	Relevante Spalten in den Dateien	18
2	Ausgabewerte UNEW	21
3	Mögliche Kombinationen für Jackknife und Datenauswahl	25
4	Parameter der QCD-Daten	27
5	Ergebnis des Gauß-Tests für SingleChain Jackknife und UNEW	28
6	Ergebnis des Gauß-Tests für mergedchains in Jackknife und UNEW mit Replicas	31
7	Parameter der zu kurzen QCD-Daten	32
8	Ergebnis des Gauß-Tests für mergedchains in Jackknife und UNEW mit Replicas mit Fehlschlag	32
9	Ergebnisse für Mean mithilfe der Jackknifemethode	36
10	Ergebnisse für Varianz mithilfe der Jackknifemethode	36
11	Ergebnisse für Skewness mithilfe der Jackknifemethode	37
12	Ergebnisse für Kurtosis mithilfe der Jackknifemethode	37
13	Ergebnisse für Mean mithilfe von UNEW	38
14	Ergebnisse für Varianz mithilfe von UNEW	38
15	Ergebnisse für Skewness mithilfe von UNEW	39
16	Ergebnisse für Kurtosis mithilfe von UNEW	39
17	Ergebnisse mit Replicas	40
18	Ergebnisse der MergedChains mithilfe des Jackknifes	40
19	Ergebnisse der MergedChains mithilfe des Jackknifes des alternativen Datensatzes	41
20	Ergebnisse mithilfe von Replicas des alternativen Datensatzes	41

Erklärung nach § 30 (12) Ordnung für den Bachelor- und dem Masterstudien-
gang

Hiermit erkläre ich, dass ich die Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst habe. Alle Stellen der Arbeit, die wörtlich oder sinngemäß aus Veröffentlichungen oder aus anderen fremden Texten entnommen wurden, sind von mir als solche kenntlich gemacht worden. Ferner erkläre ich, dass die Arbeit nicht - auch nicht auszugsweise - für eine andere Prüfung verwendet wurde.

Frankfurt, den 12.10.2017

Maximilian März