Three Pion Scattering at Maximal Isospin from Lattice QCD at Physical Mass

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Universität Bonn

vorgelegt von
Martin Ueding
aus
Bonn

Bonn, 27.08.2020

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

Gutachter

1. Professor Dr. Carsten Urbach
   Helmholtz-Institut für Strahlen- und Kernphysik, Uni Bonn
2. Professor Dr. Thomas Luu
   Institut für Kernphysik, Forschungszentrum Jülich
   Helmholtz-Institut für Strahlen- und Kernphysik, Uni Bonn
3. Professor Dr. Bernhard Ketzer
   Helmholtz-Institut für Strahlen- und Kernphysik, Uni Bonn
4. Professorin Dr. Barbara Kirchner
   Institut für Physikalische und Theoretische Chemie, Uni Bonn

Tag der Prüfung
26.11.2020

Erscheinungsjahr
2021

Abstract

In this work the scattering of two pions in $I = 2$ and three pions in $I = 3$ is analyzed with lattice QCD. The process is computed on three Wilson clover twisted mass ensembles generated by the ETMC with total momentum up to $\boldsymbol{P}^2 \leq 4$ in all irreducible representations. One ensemble has been simulated at the physical pion mass.

The correlation functions are treated with methods to suppress thermal states and excited states. Energy extraction is performed with multiple different fit methods and a combination afterwards such that the systematic errors from fit range selection are taken into account.

The three-pion spectrum has been fitted using the RFT approach. We obtain $a_0 M_\pi = 0.0481(86)$ for the two-pion scattering length at the physical pion mass. In the three-pion sector a constant and linear coefficient of the contact interaction has been extracted. The constant term agrees both with an existing analysis and leading order ChPT, the linear term has tension with the leading order ChPT prediction at high pion masses.

Contents

1. Introduction

In classical mechanics, one can solve the two-particle problem analytically and exactly. The insight is to transform it back to a single-particle problem and have one particle move in the potential of the other. This way one can study a planet around the sun or an electron around a hydrogen nucleus. As soon as one adds a third particle, no exact analytic solution to the dynamics is known. Numerically these problems can be solved to arbitrary precision given sufficient computing resources.

The strong force—which governs the interactions of quarks and nuclei and is described by quantum chromodynamics—inherently is an infinite-particle theory at low energies. One can partition these into select few *valence quarks* and infinitely many *sea quarks*. The infinitely many sea quarks are dealt with in the established frameworks. Too many bunches of valence quarks (*hadrons*) still pose a problem to the indirect interaction determination that has to be used in the simulations, the *Lüscher method*. For quite a while only the interaction of two hadrons could be described. Pions are the lightest hadrons and can only be created in pairs from the vacuum due to G-parity. The dynamics could only be described to just below $4M_\pi$. This restriction is rather severe for simulations at physical pion masses, most hadronic resonances of interest lie above this threshold.

Additionally, many hadronic resonances like the Roper resonance [1] decay into more than two particles. Besides the decay channel into $\mathrm{N}\pi$, it also dacays into $\mathrm{N}\pi\pi$. The latter can only be interpreted in with a three-particle formalism. The mass of the Roper resonance is at odds with the current understanding of the quark model, a lattice study from first principles would be highly desirable. Another example for a decay into more than two particles is the ω-meson with the largest branching fraction into three pions in p-wave [2]. A lattice investigation of the X, Y and Z resonances also depends on the existence of a suitable multi-particle formalism.

Recent developments have extended the Lüscher formalism to three particles. The relevant threshold has been lifted to $5M_\pi$ and it is hoped that this will only be the start to extend it to many more particles. In this work we will use these relatively new theoretical techniques and apply them to lattice simulation data of three pion scattering. In this way the formalism can be evaluated on a simple system and vetted for more complicated scattering channels.

The main part of this work is the determination of the three-pion interacting energies on three Wilson clover twisted mass lattices generated by the Extended Twisted Mass Collaboration, one of them with physical pion mass. The energy levels are extracted for moving frames up to $\boldsymbol{P}^2 \leq 4$ using several operators to probe the spectrum in each irreducible representation. Careful treatment of excited states and thermal pollution allows for an estimation of the systematic errors in the energy extraction.

Fitting these energies with an appropriate three-pion interaction model allows the quantification of a three-pion contact interaction strength. Additionally we provide a determination of the two-pion scattering length $a_0 M_\pi = 0.0481(86)$ at the physical point, which is compatible with previous results [3]. The three-pion interaction is parametrized linearly, the constant term agrees with an independent determination as well as leading order chiral perturbation theory (ChPT). The linear term is at odds with leading order ChPT at high pion masses.

In order to obtain these physical results, a lot of software development is needed. In the last chapter we will cover a few of the important changes to existing software packages and a new analysis framework that provides much flexibility as well as a stringent workflow organization.

2. Lattice QCD

On the level of the nucleus the dominating force comes from the strong interaction. Electromagnetism and the weak force only contribute minor corrections and therefore it is permissible to focus on this single interaction described by the theory of *quantum chromodynamics* (QCD). A general introduction to quantum field theory (QFT) and QCD can for instance be found in the textbook by Peskin and Schroeder [4]. The work presented in this thesis uses the same theoretical foundation as used in the author's master thesis [5]. A certain overlap of the theory chapters it to be expected. Like any other quantum field theory, QCD is expressed as a Lagrange density [6]:

$$L = \frac{1}{2} \operatorname*{Tr}_{\text{color}} \left(\boldsymbol{G}_{\mu\nu} \boldsymbol{G}^{\mu\nu} \right) + \bar{\boldsymbol{\psi}} \left(\mathrm{i} \not{\boldsymbol{D}} - \boldsymbol{m} \right) \boldsymbol{\psi} \,.$$

This expression contains the following quantities:

- The gauge field $\boldsymbol{A}$ is a Lorentz vector field in space-time with one four-count index μ but also a 1-form in the $\mathfrak{su}(3)$ algebra (adjoint representation) with an eight-count index a. The components are $A^a_\mu(\boldsymbol{x})$. Using the generators of the SU(3) group, the Gell-Mann matrices λ^{ij}_a one can form the $\mathfrak{su}(3)$ matrix $\boldsymbol{A}_\mu = A^a_\mu \lambda^{ij}_a$.

- $\boldsymbol{G}$ is the gluon field strength tensor built from gauge potential $\boldsymbol{A}$ as $\boldsymbol{G} = \mathrm{d}\boldsymbol{A} + \boldsymbol{A} \wedge \boldsymbol{A}$. This makes the field strength tensor $\boldsymbol{G}$ a two-form over space-time. The components are $G^{ab}_{\mu\nu}(\boldsymbol{x})$.

- The spinor field $\boldsymbol{\psi}$ describes the quark fields. It it complex Grassmann number values valued, has a three-count color index (i), a four-count spin/Lorentz index α and a flavor index f. The components are $\psi^i_{\alpha f}(\boldsymbol{x})$. Its adjoint $\bar{\psi} = \boldsymbol{\psi} \boldsymbol{\gamma}^0$ is defined with the Dirac matric $\boldsymbol{\gamma}^0$.

- $\not{D} = \gamma^\mu \mathrm{D}_\mu$ is the “slashed” covariant derivative, the contraction is done with the Dirac matrices. The covariant derivative D_μ is defined as $\mathrm{D}_\mu = \partial_\mu + \mathrm{i}A_\mu$, where ∂_μ is the usual partial derivative along direction $\boldsymbol{e}_\mu$.

- $\boldsymbol{m}$ is the quark mass matrix which acts purely in flavor space and is diagonal in all other spaces.

The first part of the action is the “gauge part”, the second the “fermionic part”. The trace operation only acts on the color indices and not in the other spaces. And in the whole of this work the constants c and $\hbar$ are set to unity, yielding the natural units of high energy physics.

Observables in the form of an operator $\hat{O}$ are extracted using the Feynman path integral [7]. All information can be extracted with correlation functions as

$$\langle 0|T\hat{O}|0\rangle = \frac{\int \mathscr{D}\boldsymbol{A}\,\mathscr{D}\bar{\boldsymbol{\psi}}\,\mathscr{D}\boldsymbol{\psi}\; O(\boldsymbol{A},\boldsymbol{\psi},\bar{\boldsymbol{\psi}})\,\exp(\mathrm{i}S(\boldsymbol{A},\boldsymbol{\psi},\bar{\boldsymbol{\psi}}))}{\int \mathscr{D}\boldsymbol{A}\,\mathscr{D}\bar{\boldsymbol{\psi}}\,\mathscr{D}\boldsymbol{\psi}\;\exp(\mathrm{i}S(\boldsymbol{A},\boldsymbol{\psi},\bar{\boldsymbol{\psi}}))}\,.$$

On the left side, there is the time ordering operator T. If $\hat{O}$ is a product of operators at different times, they will be ordered (commuted or anti-commuted) such that the times are always in increasing order. The action S is the integral over all space-time, $\int \mathrm{d}^4\boldsymbol{x}\, L$. Analytic computation of the path integral is possible when the fields enter at most quadratic in the Lagrange density, they are free fields then. As soon as an interaction is added, the integral usually cannot be solved in full. There are some particular models in certain space-time dimensionality that can be solved. However, for the interactions in the standard model, one cannot solve the path integral exactly. If the interaction is weak, like with quantum electrodynamics, the exponential function can be expanded in powers of the coupling strength. The terms of this expansion can be visualized as Feynman diagrams. For the low energy scale of QCD the interaction is large and the expansion cannot sensibly be truncated. The methods that are used instead are presented in this chapter.

2.1. Simulation

Expansion of the path integral in powers of the gauge coupling will not work in the low energy regime that the interactions of mesons and baryons are part of. We will pursue a numerical approach and solve a discretized version on large computers.

Brute force discretization of the path integral will not get us very far. The integrand is an oscillating phase and most contributions far away from the classical path will just cancel each other away. Rather one would try to use *importance sampling*, a Monte Carlo technique. It can be used to solve an integral of the form $I = \int \mathrm{d}x\, f(x)P(x)$, where the factor $P(x)$ can be interpreted as a probability density. Then one generates N values x such that they are distributed like $P(x)$, this is also called sampling from P. For all these values x_i one computes $f(x_i)$ and obtains the estimate for the integral I as

$$I \approx \frac{1}{N} \sum_i f(x_i) \,.$$

This technique has the advantage that it converges towards the result as $1/\sqrt{N}$ independent of the dimension of the space. Regular grid based approaches usually fall behind in spaces which are more than ten dimensional. As we will see later, the dimensionality involved here is significantly larger.

For the path integral the operator O is interpreted as the function f, the value x is comprised of the values of $\boldsymbol{A}$, $\boldsymbol{\psi}$ and $\bar{\boldsymbol{\psi}}$. The problem is that the complex exponential is not real, positive and bounded from above. The common procedure is the *Wick rotation* which transforms the time t to imaginary time $\tau = -\mathrm{i}t$ such that the oscillating phase factor $\exp(\mathrm{i}S)$ becomes a probability weight $\exp(-S)$ which is positive and bounded because S is bounded from below. Introducing a normalization $Z = \int \mathscr{D}\boldsymbol{A}\, \mathscr{D}\bar{\boldsymbol{\psi}}\, \mathscr{D}\boldsymbol{\psi}\, \exp(-S(\boldsymbol{A}, \boldsymbol{\psi}, \bar{\boldsymbol{\psi}}))$, one can bring the path integral into this suggestive form, with function f at front and probability P at the end:

$$\langle 0|T\hat{O}|0\rangle = \int \mathscr{D}\boldsymbol{A}\, \mathscr{D}\bar{\boldsymbol{\psi}}\, \mathscr{D}\boldsymbol{\psi}\, O(\boldsymbol{A}, \boldsymbol{\psi}, \bar{\boldsymbol{\psi}})\, \frac{\exp(-S(\boldsymbol{A}, \boldsymbol{\psi}, \bar{\boldsymbol{\psi}}))}{Z} \,.$$

Spacetime has uncountable many points, therefore a state $(\boldsymbol{A}(\boldsymbol{x}), \boldsymbol{\psi}(\boldsymbol{x}), \bar{\boldsymbol{\psi}}(\boldsymbol{x}))$ cannot be represented on a computer. A finite volume part of it needs to be discretized to have a finite number of points. The discretization used is a equidistant lattice with spacing a in all four dimensions. There are a lot of intricacies which are beyond the scope of this introduction. There are gauge links, plaquettes and pseudo fermions. For a comprehensive introduction into lattice QCD, the reader is referred to the book by Rothe [8].

With this discretization in place, the path integral becomes an integral of finite dimension which can be sampled on a computer. The number of lattice sites is still

Name	$L^3 \times T$	M_π/MeV	$M_\pi L$	Configurations
cA2.09.48	$48^3 \times 96$	134	2.98	1604
cA2.30.48	$48^3 \times 96$	242	5.37	350
cA2.60.32	$32^3 \times 64$	340	5.05	337

Table 2.1.: Ensembles used in this work.

very large, say $48^3 \cdot 96 = 10\,616\,832$ for a typical lattice used in this work. Each point then has $4 \cdot 9 = 36$ complex degrees of freedom (direction and color matrix). The high number of variables lends itself to sampling with Monte Carlo methods like the HMC [9]. An accessible introduction to the HMC for QCD is the one by Lippert [10].

Wilson [11] has introduced a gauge invariant way to discretize the Lagrangian of QCD onto the lattice. One caveat is the so-called *no-go theorem* by Nielsen and Ninomiya [12] which states that there will be one doubler fermion per dimension if the theory exhibits (a) gauge invariance, (b) translational invariance, (c) locality, and (d) chiral symmetry in the massless limit. In order to make the doublers heavy enough to decouple from the spectrum, one of the properties needs to be given up. We chose to give up the chiral symmetry by the addition of the *Wilson term* $-\boldsymbol{\psi} a \nabla_\mu^* \nabla_\mu \boldsymbol{\psi}$ to the action. This term couples left and right handed spinors, just like the mass term. It makes the doublers heavy, but also introduces additive mass renormalization.

The basic *Wilson action* exhibits lattice spacing artifacts or order $\mathrm{O}(a)$. A tool for reducing $\mathrm{O}(a)$ effects in most hadronic quantities is the *twisted mass* formulation [13, 14]. The mass and the Wilson term are the two terms which break chiral symmetry. By making them orthogonal in isospin space the additive renormalization is effectively canceled. This needs some non-perturbative tuning in order to achieve maximal twist. Using the clover term [15] one can reduce the $\mathrm{O}(a^2)$ artifacts at cost of different $\mathrm{O}(a^3)$ artifacts. The clover coefficient c_sw usually needs to be tuned non-perturbatively.

This work uses a subset of the $N_\mathrm{f} = 2$ Wilson clover twisted mass ensembles by the ETMC [16] that have already been used in many other projects, including meson scattering [17–22]. The ensembles used are listed in Table 2.1. For all ensembles we have a lattice spacing $a = 0.0914(02)(17)\,\mathrm{fm}$ and the clover coefficient $c_\mathrm{sw} = 1.57551$. They use the Iwasaki gauge [23].

In total the fermionic operator takes the form

$$\not{D}_{\text{Wilson}} + m_{\text{cr}} + \mathrm{i}\mu\gamma_5\tau_3 + \frac{\mathrm{i}}{4}c_{\text{SW}}\boldsymbol{\sigma}_{\mu\nu}\boldsymbol{G}_{\mu\nu}\,,$$

where the terms are (1) the Wilson Dirac operator, (2) mass set to critical mass, (3) twisted mass and (4) the clover term.

The ETMC also has a large group of ensembles with $N_\text{f} = 2 + 1 + 1$ Wilson twisted mass ensembles [24] which have been used in various meson scattering projects in the work group [18, 20, 25–31] and beyond. Although the 15 ensembles allow for extrapolations in pion mass, lattice spacing and lattice size, they lack a clover term and the pion masses do not include the physical one. As three-pion scattering has already been computed on unphysical pion masses [32], this work includes physical pion mass for novelty. A new group of $N_\text{f} = 2+1+1$ Wilson clover twisted mass ensembles are in the making, but was not yet available during the creation of this work.

2.2. Group theory

Symmetries play a great role in physics. When working with angular momentum on the lattice one is forced to review how it interacts in the continuum to transfer the concepts to the lattice. We will do a short review of the group theoretical concepts that are necessary for this work.

A mathematical group is a set G, a neutral element 1 and an operation $\circ$, written as $(G, 1, \circ)$. The following axioms need to hold:

Neutral element There must be an element $1 \in g$ such that for every $g \in G$ it holds that $g = 1 \circ g = g \circ 1$.

Inverse element For every $g \in G$ there must be an element denoted as $g^{-1} \in G$ such that $g^{-1} \circ g = g \circ g^{-1} = 1$.

Associativity For all elements $g, h, k \in G$ holds that $g \circ (h \circ k) = (g \circ h) \circ k$.

There also exist commutative groups where additionally one has $g \circ h = h \circ g$, but these are not the kinds of groups used in this thesis.

Examples for commutative groups are real numbers and addition, $(\mathbb{R}, 1, +)$, real numbers without zero and multiplication $(\mathbb{R} \setminus \{0\}, 1, \cdot)$. The integers $\mathbb{Z}$ form a group under addition, but do not form one under multiplication as the inverse elements are not part of the group. One examples for non-commutative groups are real square matrices with non-zero determinants and the usual matrix multiplication, $(\boldsymbol{M}_{n\times n}, 1_{n\times n}, \cdot)$.

A good set of lecture notes on group theory is the one by Lüdeling [33].

2.2.1. Representation theory

The expansion into partial waves hinges on the spherically symmetric potential. From Noether's theorem we know that there must be an associated conserved quantum number: the total angular momentum. The theorem alone does not give us quantization, for this we need representation theory of symmetry groups. Every symmetry group can be represented on a vector space via matrices $D^\Gamma(g)$, where Γ labels the representation and g is a group element. The representation is a map from the group G to the matrices in the vector space V, so $G \to V \times V$. Given a group operation $\circ$ and a matrix multiplication $\cdot$ the representation must fulfill the condition

$$\forall g, h \in G: \quad D^\Gamma(g) \cdot D^\Gamma(h) = D^\Gamma(g \circ h) \,. \tag{2.1}$$

The map D does not need to be injective, it would be called *faithful* if it was.

One well known representation is the *fundamental* one. The vector space V is Cartesian space $\mathbf{R}^3$. For rotations around the z-axis the matrices take the form

$$D^{\text{fun}}(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} .$$

This representation is faithful. The name "fundamental" comes from the historic development. This form was first, then the actual group structure and representation theory was formed later. It is beneficial to think of it as just one of many possible representations.

Another often seen representation is the *trivial* one, named A_1^+. It just has $D^{A_1^+}(g) = 1$ for all group elements g. It fulfills Equation (2.1). It might seem useless at this very moment but will become clear shortly. It is not forbidden to just tack on additional dimensions to the vector space and just insert a unit matrix block to the representation matrices. As long as the extra dimensions are decoupled, they will not change the original representation. Decoupling is guaranteed when the representation matrices are block diagonal. Looking at the fundamental representation above it seems to be block diagonal with a two-dimensional representation and the trivial representation A_1^+ tacked on. This comes from the restriction to rotations around the z-axis and restricting it to the SO(2) subgroup, we will see subgroups again later.

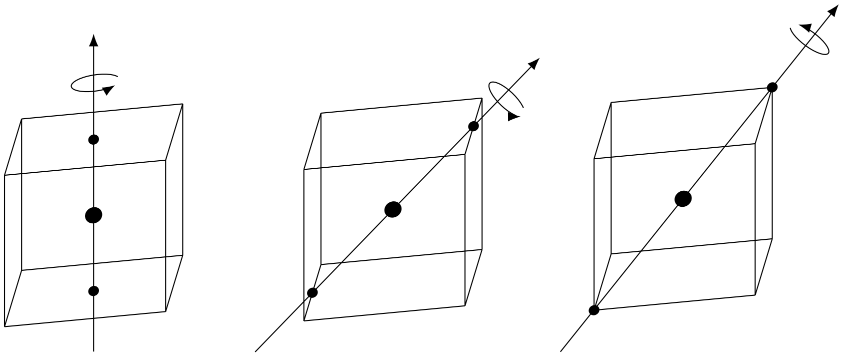
There are infinitely many representations possible for every group, but by a change of basis in V one can block-diagonalize them. The blocks are then independent representations of the same group which cannot be further reduced, they are called *irreducible representations* or *irreps* for short. Continuous groups have infinitely many irreps, finite groups only a finite number of them.

These irreps contain the information about the degeneracy of quantum states. For the SO(3) groups these irreps are well known, perhaps not under that name. The irreps are called the "angular momentum l irreps", the associated vector spaces are spanned by the spherical harmonics Y_{lm} with fixed l and have dimension $2l + 1$. From the level splitting in the hydrogen atom immersed in a magnetic field we know that the number of generate states with the same l actually is the same $2l + 1$. The conserved quantity is the label of the irrep. All states within the same irrep are degenerate and have the same energy. Breaking the full rotational symmetry with an external magnetic field restricts the group and leads to different irreps, just as has been seen in the above example of rotations around the z-axis.

2.2.2. Octahedral group

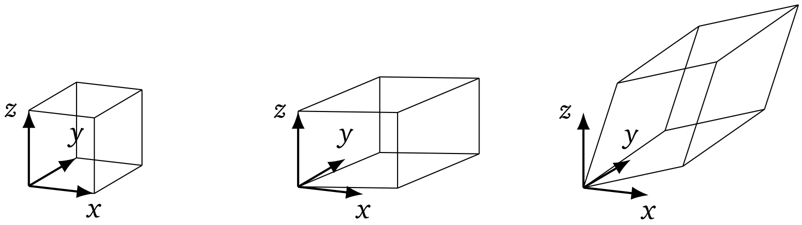
In lattice simulations the continuous rotational symmetry is broken down to the rotational symmetries of a cube. The remaining symmetry group is the octahedral group O. The cube and octahedron have the same symmetries as one solid can be obtained by using the face centers of the other as vertices. From the infinitely many rotations of a sphere only a handful of discrete rotations remain, they are exemplified in Figure 2.1.

These rotations, together with reflections, fall into 10 conjugacy classes and therefore by Laplace's theorem there are also 10 irreps for the whole group. Thinking of angular momentum in terms of irreps of the pertinent symmetry group, there are only 10 states of "angular momentum" on the lattice. The finite number of irreps on the lattice unfortunately means that the correspondence between continuum angular momentum and its lattice conterpart is not bijective. The mapping from continuum to lattice is called *subduction* and each of the lattice irreps corresponds



(a) Edge parallel (b) Face diagonal (c) Space diagonal

Figure 2.1.: Rotation axes of a cube.



(a) $(0,0,0)$ (b) $(1,1,0)$ (c) $(1,1,1)$

Figure 2.2.: Deformation of a cube by boosts along with a certain momentum $\boldsymbol{P}$.

to an infinite tower of continuum irreps. The irreps relevant for this work are listed Table 2.2.

2.2.3. Stabilizers

When particles are at zero total momentum, the center of mass frame coincides with the lattice frame. Non-zero total momentum states are needed for a well probed energy spectrum. A boost will deform the lattice and further reduce the symmetry. Figure 2.2 shows these deformations for boosts in more than one direction component. Only the group elements $g \in O$ that do not alter a total momentum $\boldsymbol{P}$ are left in a boosted lattice. This subgroup is called *stabilizer* as it stabilizes certain points or directions in space. The concept is also known as *little group* because it contains a subset of the original group.

$\boldsymbol{P}^2$	Λ^P	J^P
0	$A_1^\pm$	$0^\pm, 4^\pm, \dots$
	$E^\pm$	$2^\pm, 4^\pm, \dots$
	$T_2^\pm$	$2^\pm, 3^\pm, 4^\pm, \dots$
1, 4	A_1	$0^+, 4, \dots$
	A_2	$0^-, 4, \dots$
	B_1	$2, \dots$
	B_2	$2, \dots$
	E_2	$1, 3, \dots$
2	A_1	$0^+, 2, 4, \dots$
	A_2	$0^-, 2, 4, \dots$
	B_1	$1, 3, \dots$
	B_2	$1, 3, \dots$
3	A_1	$0^+, 3, \dots$
	A_2	$0^-, 3, \dots$
	E_2	$1, 2, 4, \dots$

Table 2.2.: Coupling of the lattice irreps to the continuum angular momenta and parity. [34, Table 2]

In the example with the fundamental representation of SO(3) for rotations only around the z-axis such a stabilizer was effectively considered. This SO(2) subgroup directly becomes block-diagonal and therefore reducible. In general the irreps will be different in a stabilizer group.

2.2.4. Group theoretical projection

The quantum field theoretical operators that are used to probe the gauge field need to have definite quantum numbers in order to be useful. Besides particle content, isospin and total momentum they also need to be in one of the irreps of the pertinent stabilizer group for the total momentum $\boldsymbol{P}$. Such a multi-particle operator can be constructed from single-particle operators O.

There are multiple conventions regarding the individual particle momenta $\boldsymbol{p}_i$. In this work we use a convention which uses relative momenta $\boldsymbol{q}_i$ and a total momentum $\boldsymbol{P}$ such that the individual particle momenta $\boldsymbol{p}_i$ are

$$\boldsymbol{p}_1 = \boldsymbol{P} - \boldsymbol{q}_1 - \boldsymbol{q}_2 \,, \quad \boldsymbol{p}_2 = \boldsymbol{q}_1 \,, \quad \boldsymbol{p}_3 = \boldsymbol{q}_2 \,.$$

This formulation has the advantage that conversion back and forth is rather easy easy to read off. In the case of two identical particles the formulation $\boldsymbol{p}_1 = \boldsymbol{P}/2 + \boldsymbol{q}$ and $\boldsymbol{p}_2 = \boldsymbol{P}/2 - \boldsymbol{q}$ also has its appeal [31] but is not used in this work.

For each total momentum magnitude $\boldsymbol{P}^2$ we have a *reference total momentum* denoted $\boldsymbol{P}_\text{ref}$. These are (0, 0, 0), (0, 0, 1), (1, 1, 0), (1, 1, 1) and (0, 0, 2).

One can construct a multi-particle operator with N_p particles from single particle operators $O(\boldsymbol{p})$ by building products of them and projecting them into the proper group theoretical irrep. The necessary prescription has been worked out by various people in the field and was given by Werner et al. [30] in a form very close to this one:

$$O_\Gamma^{\alpha\beta}(\boldsymbol{P}, \{\, \boldsymbol{q} \,\})^\dagger = \sum_{g \in \text{Stab}(\boldsymbol{P}_\text{ref})} D^\Gamma_{\alpha\beta}(g)^* \prod_{i=1}^{N_\text{p}} O^{J_i}_{iM_i''}(\boldsymbol{R}_{\tilde{g}}^{-1} \boldsymbol{R}_g \boldsymbol{R}_{\tilde{g}} \boldsymbol{p}_i)^\dagger \,. \tag{2.2}$$

The version given in the cited thesis also works for particles of non-zero spin, which is a generality not needed here with only pseudo-scalar particles. There is a lot to unpack in this equation. One has α and β which are the row and column indices of the irreducible representation. All matrix elements of an irrep behave the same way, it just might that a particular operator gives a zero in some elements.

There one has to simply chose a different matrix element. The matrices $\boldsymbol{D}^{\Gamma}$ are the irreducible representations. The general equation also contains the Wigner-D matrices $\boldsymbol{D}^{J}$, which are just the spin-J representations from the continuum. For $J = 0$ all elements are just unity, therefore they are omitted from the equation. The matrices $\boldsymbol{R}_g$ are the Cartesian representations of elements g.

$\tilde{g}$ is the group element which brings $\boldsymbol{P}_{\text{ref}}$ to $\boldsymbol{P}_{\text{cm}}$. Precisely, we have $\boldsymbol{P}_{\text{cm}} = \boldsymbol{R}_{\tilde{g}} \boldsymbol{P}_{\text{ref}}$. The argument of the single operator, $\boldsymbol{R}_{\tilde{g}}^{-1} \boldsymbol{R}_g \boldsymbol{R}_{\tilde{g}} \boldsymbol{p}_i$, therefore is the rotation of the single particle momentum to the reference frame, application of the group element and rotation back into the frame of interest. This way the stabilizer groups only need to be known for the reference frames.

In this work the only single-particle (annihilation) operators that is used are the charged pion operator $\pi^+ = -\bar{u}\gamma_5 d$ and its conjugated counterpart $\pi^- = \bar{d}\gamma_5 u$.

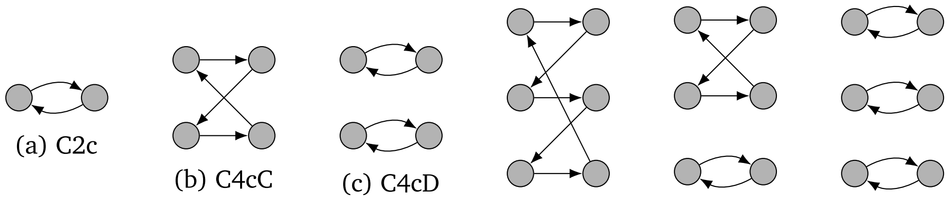
Later on for the Lüscher method we will need the *subduction coefficients* s that connect lattice and continuum symmetry. For the mesonic cases these are given by Göckeler et al. [35, Tables 10, 12 and 14]. For this work the subduction coefficients have been computed from scratch with the same representations using Equation 2.90 from Reference [31] because phase freedoms and other choices lead to incompatible coefficients if not everything is chosen consistently. The following expression is used, with a phase freedom ϕ_M.

$$s_{L,m}^{\Gamma,\alpha,n} = \sum_{m'=-L}^{L} \phi_M \frac{\dim(\Gamma)}{|\operatorname{Stab}(\boldsymbol{P}_{\text{ref}})|} \sum_{g \in \operatorname{Stab}(\boldsymbol{P}_{\text{ref}})} D_{\alpha\beta}^{\Gamma}(g)^* \, D_{mm'}^{L}(g) \,. \tag{2.3}$$

These equations are implemented in the sLapH Projections NG package [36], which is also covered in Section 6.2 in more detail.

2.3. Correlator computation

Correlators containing quark operators cannot directly be computed on the lattice as there are no Grassmann numbers on the computer. Instead one needs to perform the Wick contractions and decompose the correlator as a linear combination of diagrams. This can be illustrated with a correlation function from two pions to two



(a) C2c (b) C4cC (c) C4cD (d) C6cC (e) C6cCD (f) C6cD

Figure 2.3.: Diagrams used in this work. Only quark propagation is shown, gluon interactions are taken into account to all orders by the gauge configurations. The gray vertices on the left are on the source time slice, vertices on the right are on the sink time slice. The sLapH method removes all the spatial information, therefore permutations of the source or sink vertices yield the same correlation function.

pions, $\langle \pi^-(t)\, \pi^-(t)\, \pi^+(0)\, \pi^+(0) \rangle$. We perform one particular Wick contraction:

$$\langle \bar{\mathrm{d}}(t)\gamma_5 \mathrm{u}(t)\, \bar{\mathrm{d}}(t)\gamma_5 \mathrm{u}(t)\, \bar{\mathrm{u}}(0)\gamma_5 \mathrm{d}(0)\, \bar{\mathrm{u}}(0)\gamma_5 \mathrm{d}(0) \rangle$$

$$= \langle \gamma_5\, P_{\mathrm{u}}(t,0)\, \gamma_5\, P_{\mathrm{d}}(0,t)\, \gamma_5\, P_{\mathrm{u}}(t,0)\, \gamma_5\, P_{\mathrm{d}}(0,t) \rangle \,.$$

This has a particular topology, namely one quark-connected piece where the propagators cross back and for from source to sink time slice and back. For the scattering of three pions at maximum isospin there are six diagrams that contribute, these are illustrated in Figure 2.3. The nomenclature here works as follows. The first "C" stands for "correlator" and is always fixed, then the number of operators comes as a number. The "c" stands for conjugated (or charged) and means that half of the propagators use the γ_5 hermiticity for a flavor change in the light quark. These are needed when working with charged pions in twisted mass. The trailing letters denote the shape of the diagram. A "C" stands for "cross", a "D" is a "direct" diagram and the "CD" is a combination of both. The Wick contraction of the example above yields the "C4cC" diagram.

One can compute propagators from point (or similar) sources. These have the disadvantage that they have to be computed for each Dirac structure and momentum and are therefore not easily reusable. In order to separate propagators and the particle operators (with Dirac structure and momentum) the Laplacian Heaviside method [37] can be employed. With the method one can factorize a correlation function into tensors that either depend on the inversion or on Dirac structure and momenta such that all parts are small enough to store. These building blocks can be generated once and then used to analyze various physical scattering processes.

The first step is a lossy compression of the gauge field. One computes the eigenspectrum of the Laplacian operator on each gauge configuration to obtain the matrix of eigenvectors $\boldsymbol{V}_\triangle$ and eigenvalues $\boldsymbol{\Lambda}_\triangle$. Using a cutoff σ_s^2 one can then construct a smearing operator $\mathscr{S}$ (script S) which projects the spectrum onto the lowest eigenvalues only:

$$\mathscr{S} = \boldsymbol{V}_\triangle \, \Theta(\sigma_\mathrm{s}^2 + \boldsymbol{\Lambda}_\triangle) \, \boldsymbol{V}_\triangle^\dagger \,.$$

The matrix $\Theta(\sigma_\mathrm{s}^2 + \boldsymbol{\Lambda}_\triangle)$ is diagonal and has 0 and 1 entries on the diagonal. Therefore one can just discard the corresponding eigenvectors from $V_\triangle$ and call that $\boldsymbol{V}$. The smearing operator therefore is $\mathscr{S} = \boldsymbol{V}\boldsymbol{V}^\dagger$.

This allows the introduction of a *quark line* $\mathscr{Q}$ (script Q) that is derived from the propagator $\boldsymbol{M}^{-1}$:

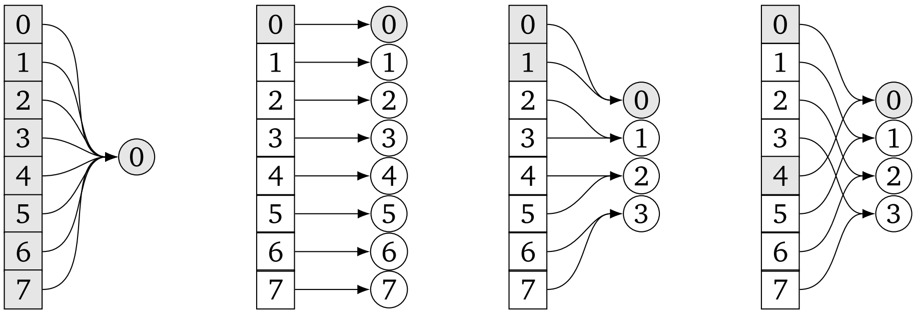
$$\mathscr{Q} = \mathscr{S}\boldsymbol{M}^{-1}\mathscr{S} = \boldsymbol{V}(\boldsymbol{V}^\dagger \boldsymbol{M}^{-1} \boldsymbol{V})\boldsymbol{V}^\dagger \,.$$

The key advantage is that the dimensions of $\boldsymbol{V}^\dagger$ are such that it has few rows (corresponding to the compressed space) and many columns (corresponding to full space and color). The object $\boldsymbol{V}^\dagger \boldsymbol{M}^{-1} \boldsymbol{V}$ therefore has a reduced size and can be stored on disk. When computing full correlation functions later on, the Dirac structures $\boldsymbol{\Gamma}$ and momenta $\boldsymbol{p}$ will enter as $\boldsymbol{V}^\dagger \exp(\mathrm{i}\boldsymbol{p}\cdot\boldsymbol{x})\boldsymbol{\Gamma}\boldsymbol{V}$, which can be computed independently of the quark line also a reduced size. Inversions have to be done only once, all possible contractions can be built up from these building blocks later on.

The required number of inversions are still very high, and given usual gauge noise it is a waste of resources to perform all of them. Therefore the *stochastic* LapH method [38] uses random noise vectors $\boldsymbol{\eta}$ with $E(\eta_i) = 0$ and $E(\eta_i \eta_j) = \delta_{ij}$. For each of these random vectors (indexed by r, N_R of them) one solves $\boldsymbol{M}\boldsymbol{X}^r = \boldsymbol{\eta}^r$. These solutions then approximate the propagator as

$$M_{ij}^{-1} \approx \frac{1}{N_\mathrm{R}} \sum_{r=1}^{N_\mathrm{R}} X_i^r \eta_j^{r*} \,.$$

One key feature of the sLapH method is the *dilution*. It can be applied in time, Dirac space and LapH space. The idea is that the random noise vectors get diluted



(a) no dilution (b) full dilution (c) block-4 (d) interlace-4

Figure 2.4.: The dilution schemes as introduced by Morningstar et al. [38]. Mapping from original space (squares) to the dilution blocks (circles). The original space has eight elements in this illustration. This figure has already been published in Reference [5].

such that certain off-diagonal elements are less noisy by a projection operator $\boldsymbol{P}^{(b)}$ where b labels the blocks. Figure 2.4 shows how the different dilution schemes can be visualized. We generally use block dilution in time, full dilution in Dirac space and interlace dilution in LapH space. For the cA2.09.48 and cA2.30.48 ensembles we have a time dilution block size of 3, cA2.60.32 has a block size of 2.

These dilution operators are then already contracted such that one has minimal storage requirements for the building blocks. The three quantities that are used in the contraction are the “V-dagger-V” objects

$$\sum_{\boldsymbol{x}} \boldsymbol{V}^\dagger \exp(\mathrm{i}\boldsymbol{p}\cdot\boldsymbol{x})\boldsymbol{\Gamma}\boldsymbol{V}\,,$$

the random vectors (RV) $\boldsymbol{\rho}^r$ and the perambulators

$$\boldsymbol{V}^\dagger \boldsymbol{M}^{-1}\boldsymbol{V}\boldsymbol{P}^{(b)}\boldsymbol{\rho}\,.$$

Using these building blocks one can compute correlation functions that are expressed in terms of the propagators. As an example we will take a look at the C20 diagram, which is just a single pion propagating assuming that the light flavors are the same. In twisted mass this involves another propagator conjugation for the flavor change, which we have omitted here for illustration. We start right after the Wick contraction, insert smearing operators, insert stochastic unit matrices in the form random vectors and recast the expression in term of the building blocks.

$$\begin{aligned}
C_\pi &= \mathrm{Tr}(\gamma_5 \boldsymbol{M}^{-1} \gamma_5 \boldsymbol{M}^{-1}) \\
&= \mathrm{Tr}(\gamma_5 \boldsymbol{V}\boldsymbol{V}^\dagger \boldsymbol{M}^{-1} \boldsymbol{V}\boldsymbol{V}^\dagger \gamma_5 \boldsymbol{V}\boldsymbol{V}^\dagger \boldsymbol{M}^{-1} \boldsymbol{V}\boldsymbol{V}^\dagger) \\
&= \mathrm{Tr}(\gamma_5 \boldsymbol{V}\boldsymbol{V}^\dagger \boldsymbol{M}^{-1} \boldsymbol{V}\boldsymbol{P}\boldsymbol{\eta}\boldsymbol{\eta}^\dagger \boldsymbol{P}^\dagger \boldsymbol{V}^\dagger \gamma_5 \boldsymbol{V}\boldsymbol{V}^\dagger \boldsymbol{M}^{-1} \boldsymbol{V}\boldsymbol{P}\boldsymbol{\rho}\boldsymbol{\rho}^\dagger \boldsymbol{P}^\dagger \boldsymbol{V}^\dagger) \\
&= \mathrm{Tr}(\underbrace{\boldsymbol{\rho}^\dagger \boldsymbol{P}^\dagger}_{\text{RV}} \underbrace{\boldsymbol{V}^\dagger \gamma_5 \boldsymbol{V}}_{V^\dagger V} \underbrace{\boldsymbol{V}^\dagger \boldsymbol{M}^{-1} \boldsymbol{V}\boldsymbol{P}\boldsymbol{\eta}}_{\text{Perambulator}} \underbrace{\boldsymbol{\eta}^\dagger \boldsymbol{P}^\dagger}_{\text{RV}} \underbrace{\boldsymbol{V}^\dagger \gamma_5 \boldsymbol{V}}_{V^\dagger V} \underbrace{\boldsymbol{V}^\dagger \boldsymbol{M}^{-1} \boldsymbol{V}\boldsymbol{P}\boldsymbol{\rho}}_{\text{Perambulator}})
\end{aligned}$$

All indices and arguments are suppressed in this example. It is important that the random vectors that are inserted are unique, otherwise this introduces unwanted correlations. More complicated diagrams can be built up in the same fashion from these building blocks.

Computation of all combinations of source and sink time slices takes a bit longer than 24 hours on a single node of JUWELS [39]. Longer wall times are not permitted for jobs. We therefore need to partition the work somehow. A straightforward solution has been to limit the source time slice with a certain remainder r such that $t_\text{so} \mod 3 = r$ is fulfilled. This way we can set $r \in \{0, 1, 2\}$ and compute the correlator $C(t_\text{so}, t_\text{si})$ in three jobs.

Just taking a third of the source time slices would normally result in data which just is a bit more noisy. The interplay with the block dilution however leads to three distinct correlators. Figure 2.5 shows the three strides for one particular correlator. One can see that it falls into three classes and the strides sample all of them. If one would only take a single of the strides, the resulting correlator shows a three-count jumping pattern. Therefore it is necessary to still do all three strides and average the results.

2.4. Error estimation with resampling

The usage of Monte Carlo integration for the path integral means that the results are not exact, but rather only statistically converging towards the true values. As only finite statistics is available, one needs to estimate these statistical errors in order to provide a trustworthy result. The method of choice for such error estimation is a Monte Carlo methods itself: *bootstrap*.

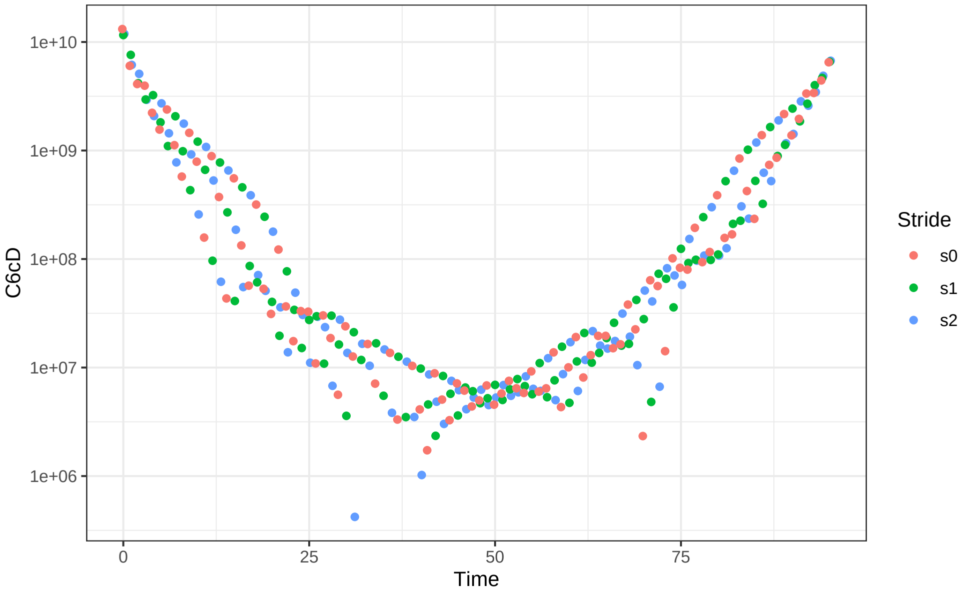


Figure 2.5.: Decomposition of the correlator into three strides. cA2b.09.48, configuration 534, 3π ground state.

Let X be a data distribution and x_i its particular measurements, of which there are N. We can use estimators like

$$\theta(X) = \frac{1}{N} \sum_{i=1}^{N} x_i$$

to compute the mean from this distribution. The uncertainty for the mean value can be computed from the standard deviation. This does not generalize to arbitrary estimators, like fitting a non-linear model to the data. Instead the bootstrap provides an error estimate for any estimator without needing to derive an error for it.

For $R \simeq 1500$ times one creates a new distribution X^{*r} by randomly drawing N elements from X with replacement. These distributions X^* are similar to the original distribution X, the deviations encode the statistical uncertainty from the finite number of measurements. For all these distributions X^* one computes the estimated values θ^{*r}. These again form the distribution Θ^*. The standard deviation of Θ^* is the statistical uncertainty associated with $\theta(X)$, the estimated value from the original data.

This procedure works with any estimator θ. Therefore one can pass the original data X and the resampled data sets X^* through all the analysis steps. At every point one can compute the uncertainty of intermediate results. There are no constraints to the analysis steps, this work uses non-linear model fitting and eigenvalue solving among many other procedures.

Slightly simpler is the *jackknife* resampling. Instead of drawing N elements with replacement, one takes all elements except the r-th one. The number of samples R is limited by the available data, in this case $R = N$. The jackknife distribution is more narrow than the bootstrap distribution, the error estimate needs to be scaled with a constant factor. The jackknife procedure is just a linear approximation to the bootstrap and has only been used as a cross-check in this work. The results are obtained with bootstrap resampling.

3. Energy extraction

The gauge configurations are probed with various operators O to yield correlation functions of the form $\langle O(t)\,O^\dagger(0)\rangle$. A model is needed before one can proceed with the extraction of energy levels. One has to explicitly write out the trace over all states. Then one uses the time evolution operator, taking the anti-periodic boundary conditions in time into account. Finally one introduces a complete set of states to extract the time dependence of the correlation function. This yields the spectral decomposition,

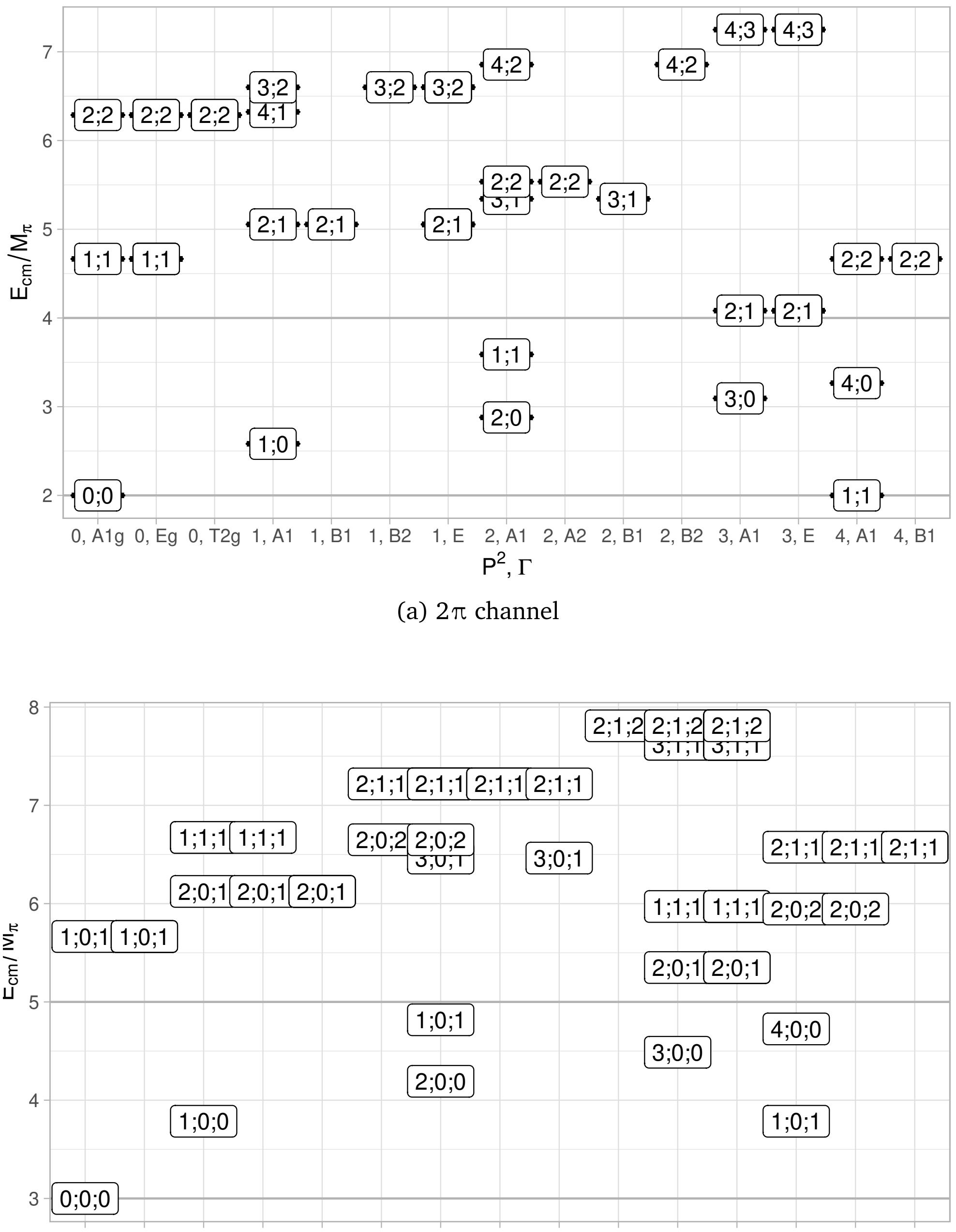
$$\begin{aligned}
\langle (O(t)\,O^\dagger(0)\rangle &= \sum_n \langle n|\,\mathrm{e}^{-H\cdot(T-t)}\,O\,\mathrm{e}^{-Ht}\,O^\dagger\,|n\rangle \\
&= \sum_m \sum_n \langle n|\,\mathrm{e}^{-H\cdot(T-t)}\,O\,|m\rangle\,\langle m|\,\mathrm{e}^{-Ht}\,O^\dagger\,|n\rangle \\
&= \sum_m \sum_n \langle n|\,O\,|m\rangle\,\langle m|\,O^\dagger\,|n\rangle\,\mathrm{e}^{-E_n\cdot(T-t)}\mathrm{e}^{-E_m t}\,.
\end{aligned}$$

By explicitly performing the exchange of m and n, one can see the cosh-like states that build the spectrum:

$$\sum_m \sum_{n\geq m} \left|\langle m|\,O^\dagger\,|n\rangle\right|^2 \exp\left(-(E_n+E_m)\frac{T}{2}\right)\cosh\left((E_m-E_n)\left(t-\frac{T}{2}\right)\right).$$

The terms are suppressed by different global factors, the higher the energies, the less they contribute to the spectrum. And larger energy difference yield a steeper slope in the cosh. In the following we will introduce methods to extract these energies from the time dependent part of the correlation function.

Determining the pion mass is easy as it is the lowest state involving only a single particle operator. Also the precision usually is way higher than any other energy extracted from more complicated systems. Using the single particle energy one can take the group theoretical prescriptions and predict the non-interacting energy levels for the given ensemble, an example is shown in Figure 3.1.



(a) 2π channel

(b) 3π channel

Figure 3.1.: Non-interacting energy levels expected on cA2.09.48 for all relevant momenta P^2 and irreps Γ. The respective elastic thresholds of $4M_\pi$ and $5M_\pi$ are denoted with a horizontal line. The numbers denote the integer momenta squared of the individual particles, $p_1^2; p_2^2$ and $p_1^2; p_2^2; p_3^2$ for two and three particles.

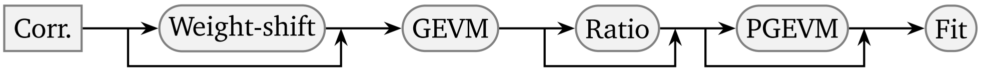


Figure 3.2.: Treatment of the correlator matrix until the energy is fitted. Some operations can be skipped. This figure has already been published [40].

Depending on the spatial lattice extent L and the pion mass M_π there are not many energy levels below the respective elastic thresholds of $4M_\pi$ and $5M_\pi$ in the 2π and 3π channels. The cA2.09.48 is the one with the fewest states below threshold as it has a fairly small $M_\pi L$.

This chapter covers the intricacies of extracting the energies from the correlators that have been computed for this project. Difficulties arise at large times slices from noise and thermal states as well as at small times from excited states. Both have treatments which will be introduced and exemplified in what follows. Figure 3.2 contains a flow chart with the methods that will be introduced. They have a particular order and can be combined.

3.1. Generalized Eigenvalue Method (GEVM)

Via the spectral decomposition it can be seen that the correlation function would be dominated by the lowest state ($m = 1$, $n = 0$) in the continuum ($T \to \infty$). As the lattice simulation is bound to work with finite temporal extent T, there is a whole tower of states to be analyzed. The back propagating terms have to be included, and the largest time separation possible is $t = T/2$.

Ideally the overlap between operator and ground state would be large, all other overlaps small; then the resulting signal would be dominated by the ground state and E_0 could be determined with small statistical uncertainty. In general one does not know which interpolating operators O couple to the various states $|n\rangle$ of the system. The established procedure is the *variational method* in the form of a *generalized eigenvalue problem* (GEVP) [41]. Instead of trying to find the single perfect operator one uses multiple ones. These have different overlaps with the states in the system. By using all cross combinations in correlation functions one can then diagonalize the correlator matrix and extract the ground state better.

Let there be different interpolating operators O_i that all have the same overall quantum numbers like total momentum, lattice irrep and parity. They could differ in the Dirac structure, have different relative momenta in a multi-particle operator, be a resonance or contain derivative operators. Using these one forms a matrix of correlation functions, $C_{ij}(t) = \langle O_i(t) O_j^\dagger(0) \rangle$. The generalized eigenvalues are defined via the condition

$$C(t)\, v(t, t_0) = \lambda(t, t_0)\, C(t_0)\, v(t, t_0)\,.$$

In practice one chooses a fixed t_0 and then performs a Cholesky decomposition on $C(t_0)$ to restate the problem as a normal eigenvalue problem which is then solved using regular methods.

The so called principal correlators $\lambda_i(t)$ then show exponentially decreasing behavior like a correlation function but with only one of the states from the correlator matrix contributing. States which are not resolved in the GEVP will still be present, in general these have much higher energies and therefore the desired state dominates earlier than before. The energy extraction is then performed on these principal correlators.

As each time slice t has independent eigenvalues, the labeling is not unambiguous. This problem gets worse when considering the bootstrap samples, it is not necessarily clear how the eigenvalues on the bootstrap samples relate to the ones from the central values. One refers to these issues as “sorting problems” and can also be seen as tracking problems. There are various strategies for numbering the eigenvalues:

1. A simple approach is to just sort the eigenvalues by value on each time slice. The bootstrap samples are done in the exact same way and matched to their central values. This approach is called “values” in the hadron library. It is the easiest approach and yields good results when the eigenvalues can be separated. Crossing of principal correlators is not possible with this approach.

2. Alternatively one can pick a particular time slice as a reference for the eigenvectors. Then on every other time slice one computes the scalar product of the eigenvectors with the ones on the reference time slice. As the eigenvectors correspond to the matrix element of operator and the state, they are assumed to be the same for all time slices. This model is called “vectors” and can potentially allow crossing of principal correlators. The downside is that the simple ranking by pairwise scalar products is not unique.

Instead of comparing every time slice to the time slice t_0, one can also compare to the previous time slice. This means that slight evolution of the eigenvectors can be tracked, but it can also mean that the tracking fails a little earlier on in the noise.

3. Embellishing the "vectors" method by taking all possible combinations will make it stable, but not necessarily any better.

Figure 3.3 shows the "values" and "vectors" methods in action for a reasonable average case. It is a GEVP with four by four operators going in, therefore there are four principal correlators. The colored blobs in this *violin plot* show the bootstrap distribution of the principal correlators on each time slice. The different colors are the numbers that the sorting algorithm has assigned. The violins are shifted horizontally such that they do not overlap, even though they are only defined at integer times.

The first time slices are just fine, the four eigenvalues clearly separate into four different chunks and there is no ambiguity about it. Starting around time slice 20 one can see that with both methods the distributions for the principal correlators 3 and 4 start to overlap. Around these times one can see differences between the two methods. With "value" sorting the values are ordered in a strictly monotonic fashion per sample, the "vectors" sorting allows principal correlators to have the same value. Later time slices grow increasingly problematic and eventually the principal correlator 2 will also start to mix with the higher two.

By eye one can spot sorting points, our brain is capable of detecting a pattern and follow a trend. This has not been successfully implemented for the computer. Fortunately the signal quality turns out to be sufficiently well that one does not have to solve this problem. It is sufficient to make pairwise Z-tests and stop using the data of $\lambda_i(t)$ for all times larger than the first where the bootstrap distribution of $\lambda_i(t)$ and $\lambda_{i+1}(t)$ are not separable at a significance level of say $\alpha = 5\,\%$. This usually occurs when the signal has become very noisy anyway, but the noise is of course worsened by sorting problems. With the Z-test in place one can easily stay clear of these time slices when choosing fit ranges and therefore prevent the sorting issues from propagating further in the analysis.

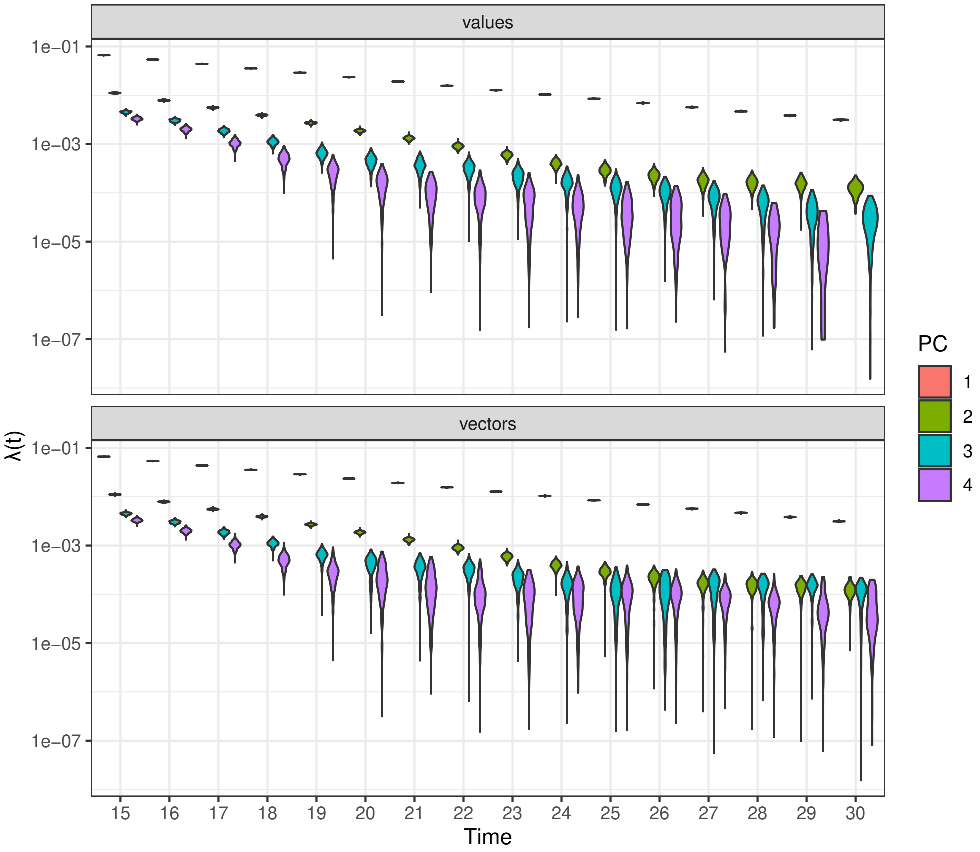


Figure 3.3.: Resampling distribution of the eigenvalues per time slice as partitioned by the GEVP sorting algorithms “values” and “vectors”. These “violins” are vertical histograms, visualizing the resampling distribution of each eigenvalue. The topmost one is so narrow that one cannot see the red fill color. Data from the cA2.09.48 ensemble, two pion channel, $d^2 = 1, A_1$ irrep, bootstrap and no thermal state removal.

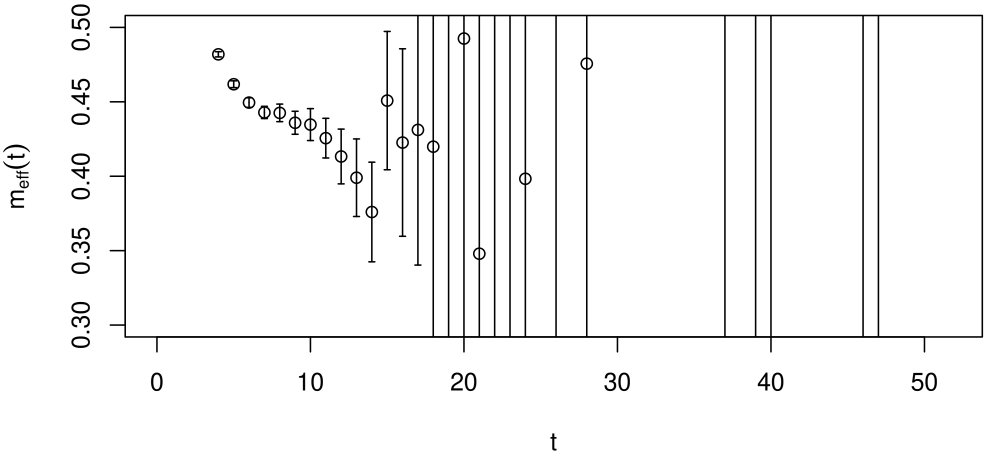


Figure 3.4.: “solve” effective mass for a case with a visible thermal pollution. cA2.09.48, 2π channel, $d^2 = 1$, B_2 irrep, first principal correlator.

3.2. Thermal state treatment

Non-trivial correlation functions often do not only contain the signal from the ground state of interest but certain other states which are called *thermal states* or *thermal pollution*. They spoil the plateau of the effective mass at large time separations. These *effective masses* will be introduced later on in more detail. At this point we only need to know that they feature a plateau when the state of interest dominates the signal. In realistic cases there will not be any plateau and the thermal states have to be treated. An example can be found in Figure 3.4 where a downwards trend completely prevents a plateau. The effective mass used here takes the back propagating part of the correlator into account, therefore it must be an additional pollution present.

We will first inspect the origin of the thermal states and use this to predict their energies for later removal. Different treatment techniques are discussed and are used for the analysis done in this work.

For the three pion interacting spectrum we probe with operators $O_\Gamma(\boldsymbol{P}, \boldsymbol{q}_1, \boldsymbol{q}_2; t)$ which transform under a given lattice irrep Γ by the construction outlined in Section 2.2.4. The particle content of this operator is $\pi^+(\boldsymbol{p}_1)\, \pi^+(\boldsymbol{p}_2)\, \pi^+(\boldsymbol{p}_3)$ such that the individual momenta map to the total and relative momenta. The computed correlators are correlation functions matrices consisting of operators of the form

$$\operatorname{Tr}\left(O_\Gamma(\boldsymbol{P}, \boldsymbol{q}_1, \boldsymbol{q}_2; t)\, O_\Gamma^\dagger(\boldsymbol{P}, \boldsymbol{q}_3, \boldsymbol{q}_4; 0)\right)$$

with the same irrep Γ and total momentum for all operators but different relative momenta at source and sink. These different source and sink operators build up the correlator matrix.

The temporal dependence of the correlation function has already been shown at the beginning of the chapter, though the explicit structure of the operator has been omitted at that point. Together with the momentum arguments we have the following expression:

$$\begin{aligned}&\operatorname{Tr}\left(O_\Gamma(\boldsymbol{P}, \boldsymbol{q}_1, \boldsymbol{q}_2; t)\, O_\Gamma^\dagger(\boldsymbol{P}, \boldsymbol{q}_3, \boldsymbol{q}_4; 0)\right) \\ &\quad = \sum_m \sum_n \langle n | \, O_\Gamma(\boldsymbol{P}, \boldsymbol{q}_1, \boldsymbol{q}_2) \, | m \rangle \, \langle m | \, O_\Gamma^\dagger(\boldsymbol{P}, \boldsymbol{q}_3, \boldsymbol{q}_4) \, | n \rangle \, \mathrm{e}^{-E_n \cdot (T-t)} \mathrm{e}^{-E_m t} \,.\end{aligned}$$

The signal that we are interested in has $n = 0$ and $m = \pi^+(\boldsymbol{p}_1)\, \pi^+(\boldsymbol{p}_2)\, \pi^+(\boldsymbol{p}_3)$. Therefore the exponential time dependence is only with $E_{3\pi}$ as the vacuum energy can be set to $E_0 = 0$. Unfortunately we also must face contributions from the case where $n = \pi^-(\boldsymbol{p}_1)$ and $m = \pi^+(\boldsymbol{p}_2)\, \pi^+(\boldsymbol{p}_3)$. The leading contribution of this kind one might expect is the one with all three momenta zero. However, for given irrep the multi particle operator does not couple to all combinations of individual momenta. The lowest lying thermal contribution therefore might be higher than one would expect. We will present an algorithm to find this one for a whole correlator matrix.

The state in between will have a matrix element $\langle \pi(\boldsymbol{p}_1) | \, O_\Gamma(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3) \, | \pi(\boldsymbol{p}_2)\, \pi(\boldsymbol{p}_3) \rangle$. The three momenta must couple to the operator O_Γ, otherwise this contribution would be non-zero anyway. Additionally the symmetry for the intermediate two pion state must be compatible with the three pion irrep Γ as well. In the continuum with spin as a conserved quantum number we can easily see this. The irreps there are the angular momentum l irreps. And as the pion itself has $l = 0$ but negative parity (0^-), we must make sure that both the O_Γ as well as the $|\pi\pi\rangle$ have the same angular momentum but opposite parity. In group theory irreps one would express this as $0^- \otimes l^+ = l^-$.

On the lattice this works similarly. The single pion on the left will always be in the A_1^- irrep, which is just the trivial irrep regarding spin and has a sign for parity. The two pion states will have positive parity in the $\boldsymbol{P} = 0$ case and therefore the three

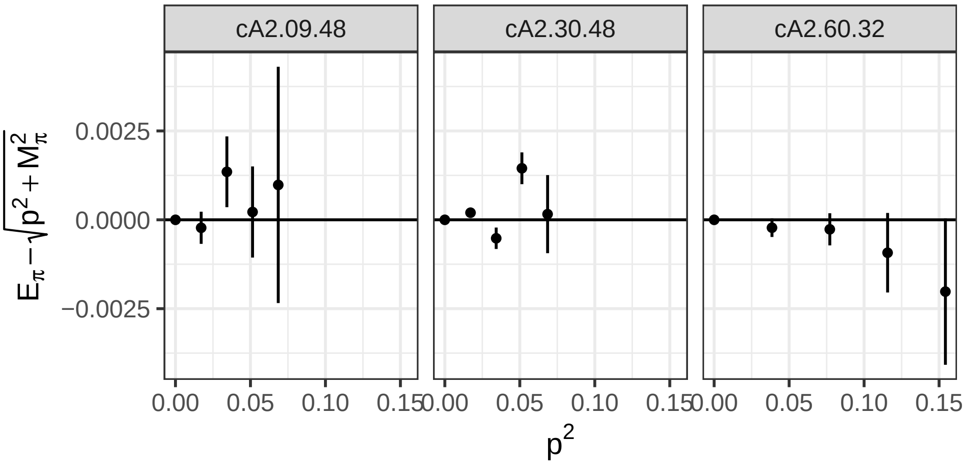


Figure 3.5.: Deviation from measured in-flight pion energy to continuum dispersion relation prediction.

pion operator O being in irrep Γ^- will need the two pions to be in Γ^+ in order for this to give a contribution as $A_1^- \otimes \Gamma^+ = \Gamma^-$ for all lattice irreps Γ. With non-zero individual momenta we have to take care that the chosen momenta actually couple to the respective irreps.

We therefore need a way to find the $E_\pi(\boldsymbol{p}_1)$ and $E_{2\pi}(\boldsymbol{p}_2, \boldsymbol{p}_3)$ that contribute to a thermal pollution to the correlator matrix with given total momentum $\boldsymbol{P}^2$ and irrep Γ^-. Once we can insert that into the spectral decomposition to find the time independent suppression factor $\exp(-(E_{2\pi} + E_\pi)T)$ and time dependent factor $\exp(-(E_{2\pi} - E_\pi)t)$ of the thermal state.

3.2.1. Choosing the energies

The $E_\pi(\boldsymbol{p}_1)$ can be sourced from the pion correlator at $\boldsymbol{P}^2 = 0$ boosted to $\boldsymbol{p}_1$ or from the pion correlator with the appropriate momentum. In general we see that the energy extracted from $\boldsymbol{P}^2 = 0$ has less statistical uncertainty than the ones with non-zero momenta. The boosted energies are verified against the energies extracted from the non-zero momentum frames, see Figure 3.5 for the deviations. As they agree within uncertainty we chose to use the boosted energies with less statistical uncertainty.

For the energy $E_{2\pi}(\boldsymbol{p}_2, \boldsymbol{p}_3)$ there are two options: Either compute the non-interacting energy from the known relative momenta or use the interacting energy measured from the correlator. Using the non-interacting energies benefits from rather low statistical uncertainties as the zero momentum pion mass can be used throughout. Also the group theory seems to be clear-cut as the single particle momenta going in are exactly known. For a weakly interacting system like the three pions at maximal isospin ($I = 3$) there is not a large difference. However, the physical issue is a two-particle state traveling in a time loop. This state consists of interacting particles and not two single particles which propagate independently. Therefore usage of the interacting energies appears to be the correct treatment and the non-interacting energies an approximation whose applicability depends on the scattering process at hand.

Most steps are the same for both kinds of energies, differences arise in Step 4 in the following procedure. Given is a total momentum $\boldsymbol{P}$ and an irrep Γ^- that describe the correlator matrix for the three pion states.

1. Find all possible combinations of individual momenta $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}$ that couple to the given irrep Γ^-. Since isospin considerations also apply it is easiest to just look at the projection prescriptions and see which relative momenta show up in the correlator matrix.

   Up to certain cutoffs for $\boldsymbol{P}^2$ and $\max_i \boldsymbol{p}_i^2 \leq 1$ and $\sum_i \boldsymbol{p}_i^2$ such that the non-interacting energies are not way beyond $5aM_\pi$ on the ensembles of interest we have a table of operators contributing to the given irreps for three pions. This data was generated by the new projection code [36]. The full tables can be found in Appendix B. An example of this algorithm with table excerpts will be given shortly.

2. Take all unique permutations (up to $N!$ for N particles) of these three momenta and also put them into the pool of momentum combinations. This means that we can just partition $\boldsymbol{p}_1$ to the one pion state and take the $\boldsymbol{p}_2$ and $\boldsymbol{p}_3$ to the two pion state without losing a combination.

3. We can always create a state $\langle \pi(\boldsymbol{p}_1)|$ in the A_1^- irrep, therefore this is trivial. We then have to check for every momentum combination if the state $|\pi(\boldsymbol{p}_2)\,\pi(\boldsymbol{p}_3)\rangle$ actually couples to the oppositve parity irrep Γ^+. If that is the case, add the current individual momentum combination to a list of *accepted* combinations.

4. The way of the energy computation differs for interacting and non-interacting energies. One needs to go through the accepted combinations and compute and do either of the two options:

   - For the non-interacting energies simply use the dispersion relation, so

   $$E_\pi^\text{free} = \sqrt{M_\pi^2 + \left(\boldsymbol{p}_1 \frac{2\pi}{L}\right)^2}$$
   $$E_{2\pi}^\text{free} = \sqrt{M_\pi^2 + \left(\boldsymbol{p}_2 \frac{2\pi}{L}\right)^2} + \sqrt{M_\pi^2 + \left(\boldsymbol{p}_3 \frac{2\pi}{L}\right)^2} \,.$$

   - The interacting energies $E_{2\pi}^\text{int}$ are extracted from a correlator matrix with a total momentum $\boldsymbol{P}'$ and irrep Γ^+. Through the variational method (GEVP) we cannot keep track of the individual momenta that went into each state. Therefore we just need to take the tower of states characterized by $\boldsymbol{P}'$ and Γ^+. For each accepted momentum combination we take $\boldsymbol{P}' = \boldsymbol{p}_2 + \boldsymbol{p}_3$ and take the lowest state from the tower of states. This gives us the $E_{2\pi}^\text{int}$ for the thermal state.

5. Compute the time independent suppression factor $\exp(-(E_{2\pi} + E_\pi)T)$. Also compute the time dependence factor $\exp(-(E_{2\pi} - E_\pi)t)$ and evaluate this at the time slice t where one expects the signal to be. Tally all the contributions in a table.

6. Pick the contribution which is deemed most obnoxious in the potential plateau region. It is not clear whether one should pick the one with the lowest suppression factor or rather the one with the strongest time dependence. We do not know the amplitudes for the different states that arise from the physical matrix elements; therefore we just set them to unity. For the end result we use the state which has the highest product of time dependent and time independent factor.

As an example for the algorithm, we can take a look at $\boldsymbol{P}^2 = 0$ in the A_1^- irrep. The three pion momenta are listed in Table 3.1. We now try all combinations of the momenta.

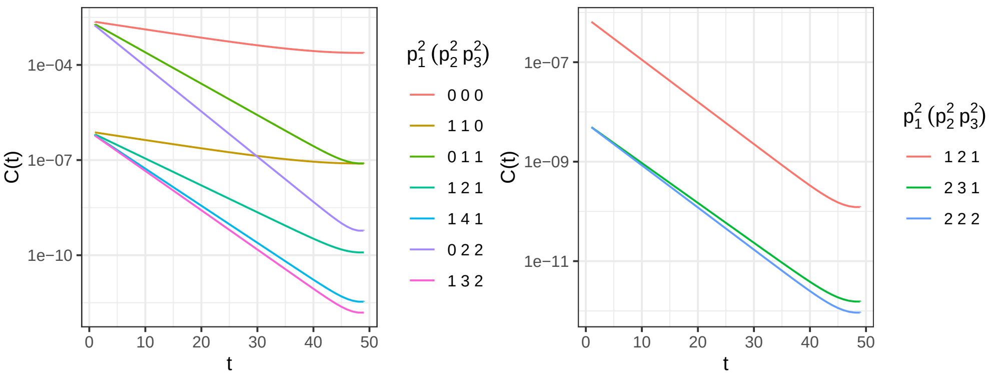
- The first row has all momenta zero, there it does not matter how one partitions the three momenta into the one and two pion system. The two pions

$\boldsymbol{P}^2$	Irrep	$\boldsymbol{p}_1$	$\boldsymbol{p}_2$	$\boldsymbol{p}_3$
0	A1u	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$
0	A1u	$(0,0,-1)$	$(0,0,0)$	$(0,0,1)$
2	B1	$(0,1,-1)$	$(0,0,1)$	$(1,0,0)$
2	B1	$(1,1,0)$	$(1,0,0)$	$(-1,0,0)$
2	B1	$(-1,1,0)$	$(1,0,0)$	$(1,0,0)$

Table 3.1.: Selection of three-pion individual momenta contributions. The full table can be found in Appendix B.

$\boldsymbol{P}^2$	Irrep	$\boldsymbol{p}_2$	$\boldsymbol{p}_3$
0	A1g	$(0,0,0)$	$(0,0,0)$
0	A1g	$(0,0,-1)$	$(0,0,1)$
0	A1g	$(-1,-1,0)$	$(1,1,0)$
1	A1	$(0,0,2)$	$(0,0,-1)$
1	A1	$(0,0,1)$	$(0,0,0)$
1	A1	$(-1,0,1)$	$(1,0,0)$
1	A1	$(-1,-1,1)$	$(1,1,0)$
1	B1	$(-1,0,1)$	$(1,0,0)$
2	B1	$(1,1,-1)$	$(0,0,1)$
4	B1	$(0,-1,1)$	$(0,1,1)$

Table 3.2.: Selection of two-pion individual momenta contributions. The full table can be found in Appendix B.



(a) $\boldsymbol{P}^2 = 0$ in A_1^-

(b) $\boldsymbol{P}^2 = 2$ in B_1

Figure 3.6.: Thermal states arising from different momentum combinations of the three pions involved for two cases on the cA2.09.48 lattice.

will have total momentum $\boldsymbol{P}^2 = 0$ and be in the A_1^+ irrep. We take a look at Table 3.2 and find that indeed there is a contribution from both momenta being zero. Therefore we add the corresponding one and two pion energies to the pool of contributions.

- In the next row we have two options. We first take the momentum zero for the single pion. The two pion system therefore still have total momentum zero, but the particles have a back-to-back momentum. We also find this in the second table, therefore this contributes as well.

- Setting $(0, 0, -1)$ away for the single particle leaves a two-pion system with total momentum $\boldsymbol{P}^2 = 1$. We find that in the fifth row of Table 3.2. Therefore this also contributes.

There are more momenta contributions possible in the three-pion operator. All thermal states that are considered are shown in Figure 3.6a. One can see that the case with all momenta zero is the most significant thermal state. We will therefore pick that one as the one to treat and ignore the others, which are suppressed by several orders of magnitude.

In the previous example all tested combinations contributed as it is the trivial irrep. We will also look at the frame $\boldsymbol{P}^2 = 2$ in the B_1 irrep as that is non-trivial and couples only to d-wave. A few momentum combinations are tried as follows.

$\boldsymbol{P}^2$	Γ	$\boldsymbol{p}_1$	$\boldsymbol{p}_2$
0	A_1^+	$(0,0,0)$	$(0,0,0)$
0	E^+	—	—
1	A_1	$(0,0,1)$	$(0,0,0)$
1	B_1	—	—
1	B_2	—	—
2	A_1	$(1,1,0)$	$(0,0,0)$
2	A_2	—	—
2	B_1	—	—
2	B_2	—	—
3	A_1	$(1,1,1)$	$(0,0,0)$
4	E	—	—
4	A_1	$(0,0,2)$	$(0,0,0)$
4	B_1	—	—

(a) Two pion channel

$\boldsymbol{P}^2$	Γ	$\boldsymbol{p}_1$	$\boldsymbol{p}_2$	$\boldsymbol{p}_3$
0	A_1^-	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$
0	E^-	$(0,0,-1)$	$(0,0,0)$	$(0,0,1)$
1	A_2	$(-1,0,1)$	$(0,0,0)$	$(1,0,0)$
1	B_2	$(-1,0,1)$	$(0,0,0)$	$(1,0,0)$
1	E	$(-1,0,1)$	$(0,0,0)$	$(1,0,0)$
2	A_1	$(1,0,-1)$	$(0,0,0)$	$(0,1,1)$
2	A_2	$(1,1,0)$	$(0,0,0)$	$(0,0,0)$
2	B_1	$(0,1,-1)$	$(0,0,1)$	$(1,0,0)$
2	B_2	$(1,1,-1)$	$(0,0,0)$	$(0,0,1)$
3	A_1	$(1,0,-1)$	$(0,0,1)$	$(0,1,1)$
3	A_2	$(1,1,0)$	$(0,0,0)$	$(0,0,1)$
3	E	$(1,1,0)$	$(0,0,0)$	$(0,0,1)$
4	A_2	$(0,-1,1)$	$(0,0,0)$	$(0,1,1)$
4	B_2	$(-1,0,1)$	$(0,0,1)$	$(1,0,0)$
4	E	$(-1,0,1)$	$(0,0,1)$	$(1,0,0)$

(b) Three pion channel

Table 3.3.: Leading thermal state contributions in the respective channels on the cA2.09.48 ensemble. The other ensembles have slightly different time dependencies on the thermal states and can lead to a different leading state.

- First we take take $(0,1,-1)$ for the single pion and have $\boldsymbol{p}_2 + \boldsymbol{p}_3 = (1,0,1)$ for the two-pion system. This has total momentum $\boldsymbol{P}^2 = 2$ but the momenta do not couple in this way. Therefore this is not a valid contribution.

- Taking $(1,0,0)$ for the single pion leaves $(0,1,0)$ as the total momentum for the two-pion system. And indeed we find this case in the table, albeit rotated under a meaningless global rotation. Therefore this contributes.

The resulting thermal states of this example are depicted in Figure 3.6b. One can see that there are way less thermal states in this irrep. The E irreps do not even show any thermal states. These results are also listed in Table 3.3, where all frames and irreps can be found.

3.2.2. Energy extraction in presence of thermal states

Given the pairs of $E_{\pi}(\boldsymbol{p}_1)$ and $E_{2\pi}(\boldsymbol{p}_2, \boldsymbol{p}_3)$ the thermal state can be predicted. Its time dependence is given by $\exp(-(E_{2\pi} - E_{\pi})t)$ in the forward term. We introduce $\Delta E := E_{2\pi} - E_{\pi}$ and write this as $\exp(-\Delta E\, t)$. There is an additional backward propagating part as well which goes as $\exp(-\Delta E \cdot (T - t))$. Together they either form a cosh (sum) or a sinh (difference). For the three pions we only have time-even operators and therefore everything will have a cosh-shape.

The ground state signal of the correlation function has a cosh-shape and therefore is modeled by

$$C(t) = A_0 \left[\exp(-E_0 t) + \exp(-E_0 \cdot (T - t))\right] .$$

When we take one thermal state into account we have the model expression

$$\begin{aligned} C(t) = A_0 &\left[\exp(-E_0 t) + \exp(-E_0 \cdot (T - t))\right] \\ &+ A_1 \left[\exp(-\Delta E\, t) + \exp(-\Delta E \cdot (T - t))\right] . \end{aligned}$$

There are two methods of extracting the energies using the knowledge of the thermal states. One is called *weight-shift-reweight* [34], the other one explicitly fits the thermal states. For both methods we need to know the ΔE.

The weight-shift-reweight method works as follows:

1. Divide the whole correlation function by the time dependence of the thermal pollution (the bracket with amplitude A_1). This way the correlation function

   $$C_\mathrm{w}(t) := C(t) / \left[\exp(-\Delta E\, t) + \exp(-\Delta E \cdot (T - t))\right]$$

   loses the time dependence in the thermal pollution and just has A_1 as an additive constant.

2. Perform a *shift* of the correlation function by δt. The weight-shifted correlation function therefore is

   $$C_\mathrm{ws}(t) := C_\mathrm{w}(t - \delta t) - C_\mathrm{w}(t) .$$

   The additive constant will drop out in this process.

3. One could fit the correlator now as is and have more complicated model for it. When the back-propagating part is neglected in both the signal and the thermal state one still can use a simple “shifted” model for this correlator and will obtain an energy that differs by ΔE from the actual signal energy.

   Instead we perform the *reweight* and multiply the weight-shifted correlator with the time dependence of the thermal state again to undo the weighting:

$$C_{\mathrm{wsr}}(t) := C_{\mathrm{ws}}(t) \cdot [\exp(-\Delta E \cdot (t - \delta t)) + \exp(-\Delta E \cdot (T - (t - \delta t)))] \,.$$

We introduce a weight factor $w := \exp(-\Delta E)$ into the notation and write the full fit model for a general cosh/sinh sign s as

$$\begin{aligned}
&\frac{1}{w^{2t} + s w^{T}} A_0 \exp(-E_0 \cdot t + T) w^{-\delta t} \\
&\cdot \Big[-[\exp(2E_0 T) + s \exp(E_0 \cdot (2t + T))]\, w^{2t} \\
&- s\, [\exp(2E_0 T) + s \exp(E_0 \cdot (2t + T))]\, w^{2\delta t - T} \\
&\quad + [\exp(E_0 \cdot (\delta t + 2T)) + s \exp(E_0 \cdot (-\delta t + 2t + T))]\, w^{\delta t} \left[w^{2t} + s w^{T} \right] \Big] .
\end{aligned}$$

We see that ignoring the back-propagating part by setting $s = 0$ the model drastically simplifies to a simple “shifted” model. The advantage of this method is that the A_1 amplitude does not need to be known or determined. Yet we have the shift and this introduces much more noise and makes finding plateaus harder.

Explicitly fitting the thermal state is simpler conceptually. One needs to use the model with the explicit thermal state and has the amplitude A_1 as an additional fit parameter. One could fix the ΔE, but rather we also introduce it as a fit parameter constrained with our prior knowledge from the considerations above. An additional summand with a normal distribution is added as a prior to the χ^2 function:

$$\exp\left(\frac{\Delta E - \overline{\Delta E}}{s(\Delta E)} \right) ,$$

where we have introduced the mean value $\overline{\Delta E}$ and its standard error $s(\Delta E)$ as extracted before. This additional prior will constrain the parameter space such that we only have three truly free fit parameters instead of four.

3.3. Fit range selection

Fitting a model directly to all available data points of the correlator does not work, unfortunately. For small times there are excited states with unknown masses. Therefore the model becomes only valid starting from a certain t_1. Additionally in most non-trivial correlation functions there is exponential error growth towards the end. For all times beyond a t_2 the signal-to-noise ratio is just too bad to fit the model. These time slices add little to the χ^2 but add another degree of freedom. Additionally possibly present thermal pollutions becomes significant towards the end. One needs to choose t_2 sufficiently small that all thermal pollutions drown in the noise.

A good starting point for choosing t_1 and t_2 (the *fit range*) is the effective mass. There are different variants depending on the expected model. The simplest approach is the "log" effective mass. It assumes that the correlator is just the forward part, $C(t) = A \exp(-Et))$, and applies

$$m_\text{eff}(t) = -\log\left(\frac{C(t)}{C(t+1)}\right) .$$

In order to include the backward propagating part one can use the "solve" method. It uses the model with forward and backward part and solves

$$\frac{C(t+1)}{C(t)} = \frac{\cosh(-m_\text{eff}(t) \cdot (t+1))}{\cosh(-m_\text{eff}(t) \cdot t)}$$

for the effective mass on each time slice numerically. The weight-shift-reweight treatment also changes the model that needs to assumed for the signal. This leads to a rather complicated effective mass, which is more complicated but in principle similar to the one above. It is called the "weighted" effective mass. Depending on the treatment of the correlators we will use the appropriate effective mass definition.

Figure 3.7 shows the effective mass for a well-behaved case from the analysis in the upper left. It also shows a zoomed version in the lower left. The upper right is the correlator with the chosen fit range and fit marked. Looking at the figure one can rather easily read off that the excited states have died down below the noise level at $t = 10$. And the noise level becomes too large after $t = 25$. Here the choice of the fit range seems rather straightforward.

As the effective mass uses derivatives of the correlator to get rid of the amplitude,

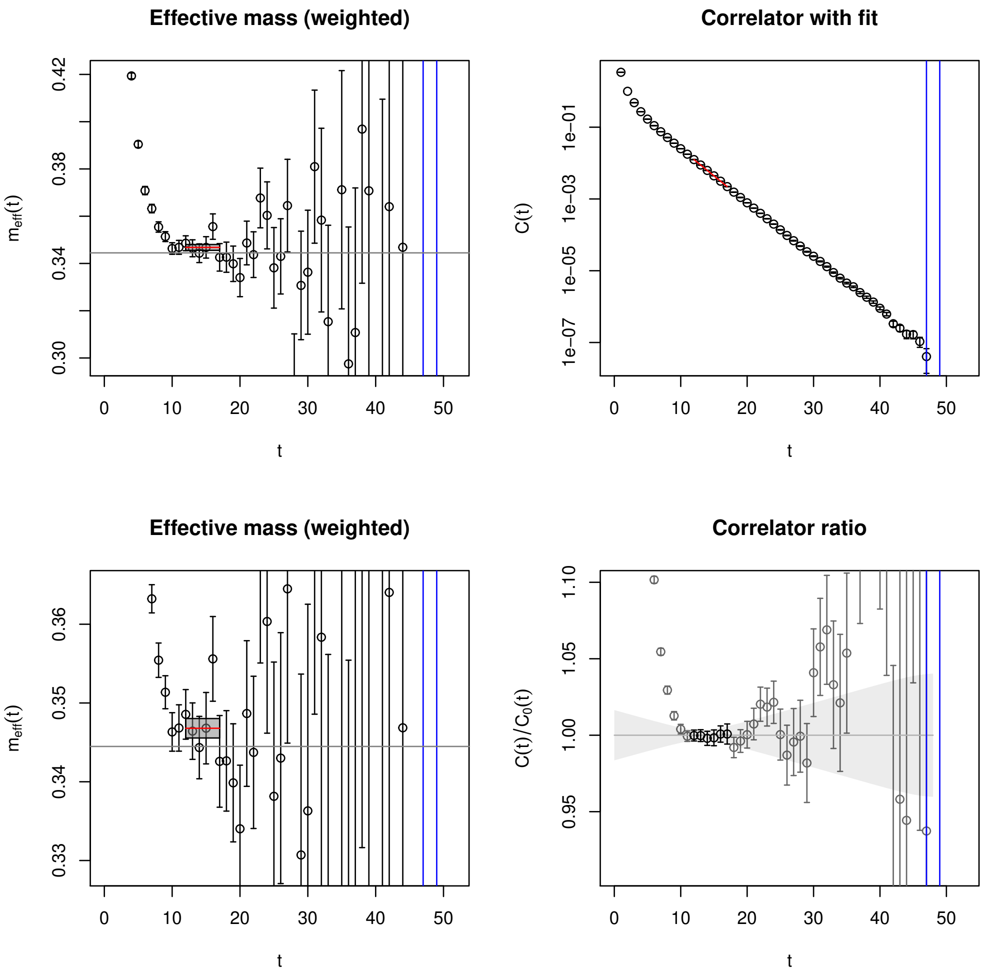


Figure 3.7.: Effective mass, correlator and fit ratio in the "weighted" model for the weight-shift-reweight correlator of the cA2.30.48 ensemble, 2π channel, $d^2 = 0$ and E irrep, first principal correlator. Blue lines denote the beginning of the GEVP sorting problem. The horizonal gray line denotes the non-interacting energy expectation.

each point in the effective mass will use two or three points of the correlator. When fitting the correlator, no such thing happens and every point is independently described by the model. To judge the quality of a fit on a given range it is better to look at the "ratio plot" which depicts $C(t)/C_{\text{model}}(t)$. This way one can see which points lie above or below the model curve and whether there is an upwards or downwards trend indicating an incorrectly estimated energy for the correlator.

In the effective mass it appears as if we could start fitting two time slices before and could stop two slices earlier to avoid the point that is significantly higher. However, the ratio plot is much better suited to determine that. Only the black points are used in the fit, the gray points are not used. As one can see that the fit range could only start one time slice earlier. This has not been included as its error is rather small and would have dominated the fit. And the fit range ends such that all the points in the fit nicely lie on the unit line in the ratio. Also the outlier is gone, that might be an artifact from taking the derivative of the correlator. Therefore the effective mass should be an initial guidance, but the ratio should be used to commit to the ranges.

In more complicated cases, however, the choice becomes rather subjective. This can be seen for a different case in Figure 3.8. The plateau cannot be easily found and one could argue that the fit range should rather be from 15 to 20 instead of the one chosen. In the effective mass each point of the plateau is compatible with the fitted value, and the plateau is not too short; this makes it a decent choice. Also in the ratio one can see that points go above and below the unit line, yet still within errors and to both sides of the line. Additionally the result is compatible with the other methods, therefore it was determined to accept this fit range. Still, one needs to find a procedure for this.

In general, discussing fit ranges with colleagues with almost always lead to disagreement. It is a problem that has bogged the field for decades and there have been various ideas to remove the subjectivity from the selection.

- Instead of doing a single fit, one could do all possible ones and average them with some procedure. This has been persued by this group for a while. The crucial ingredient is the weight that is assigned to each fit. We have used a p-value weighting. In a fit it gives the conditional probability of finding such or more noisy data *if* the model describes the data. A very low p-value therefore is interpreted as a bad model for the data. Assigning a low weight to fits with low p-value, say below 5 %, will therefore remove unsuitable fit ranges from the average. A very large p-value, say above 90 %, usually

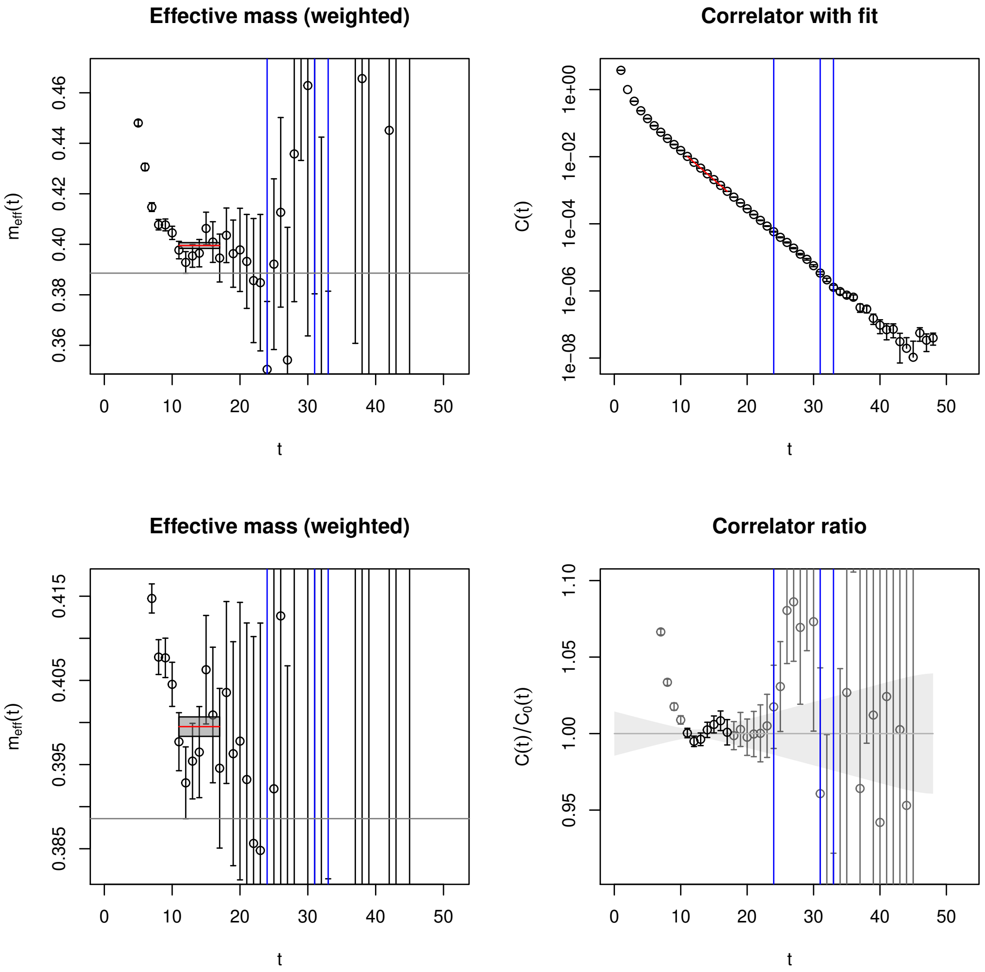


Figure 3.8.: Effective mass, correlator and fit ratio in the "weighted" model for the weight-shift-reweight correlator of the cA2.30.48 ensemble, 2π channel, $d^2 = 1$ and A_1 irrep, second principal correlator. Blue lines denote the beginning of the GEVP sorting problem. The horizonal gray line denotes the non-interacting energy expectation.

means that one has fitted noise and that everything would be compatible with that model. These would be fit ranges that just take a few points of noise at the end.

It seemed that assigning the maximum weight to $p = 0.5$ and having it fall off linearly towards $p = 1.0$ and $p = 0.0$ would favor the fits that are just right. Taking all possible fits would eliminate all subjectivity. However, most bad fits can easily be tuned to have a p-value of around 0.5. If the value is too low, just extend t_2 until sufficient noise brings down the χ^2/dof and p-value up to 0.5. If it needs to be lowered, just shorten t_2 or extend t_1 into the excited states such that the p-value comes down to 0.5.

This method had been used for a while until it was realized that favoring p-values of 0.5 does not make sense. If the model describes the data, the p-values are uniformly distributed between 0.0 and 1.0. Therefore it can only be used to discard a null hypotheses, but not to differentiate between slightly different good null hypotheses.

- For the analysis in the ρ-channel [30] the fits have been performed for the correlators with and without weight-shift-reweight independently. This gave two independent energy extractions from different looking correlators. The two fit ranges chosen of course contain subjectivity, yet the spread between them gave an estimate for the systematric error from the fit range choice.

- Matthias Fischer has implemented an improved version of the fit range weighting in his master's thesis [42]. Using Markov chain Monte Carlo methods he samples all fit ranges and possible energies in a Bayesian approach. Instead of just trying to describe the data within the fit range the model also describes the excited states. As their nature is not known, a sleight of hand was used: All points before t_1 contribute exactly 1.0 to the χ^2 and are taken to be perfectly described. This made the beginning of the fit range much more stable.

  This method is almost exactly what we were looking for, with a crucial caveat. In the process all bootstrap samples are used holisticly, the end result is an energy distribution stemming from statistical (gauge noise) and systematic (fit range choice) fluctuations. The correlation with the observation is gone. This means that fitting to the single pion correlators and then to interacting two or three pion correlators yield uncorrelated results, making energy differences less precise than would be possible with correlation taken into account. So although this solves most problems, it is not workable for computing energy shifts like in this work here.

One alternative approach might be to fit correlator ratios such that the extracted energy directly is the energy shift. Due to the thermal pollution in both the two-pion and three-pion correlators the ratio is expected to be rather noisy and hard to fit, therefore this has not been tried yet.

Additionally there is insufficient experience with this method, it is not clear how well it actually handles the systematic effects and how the results should be interpreted.

- Training a simple neural network using chosen fit ranges to predict other fit ranges does not seem to work [43]. And even if it would, systematic errors would still needed to be discussed.

For this work it the method of Reference [30] has been extended to more methods. Using each of the methods on every energy level gives a handful of different energy determinations. The resulting energies are compared and significant deviations from other energies are investigated. The choice of the fit range selection still suffers from subjectivity, but overall the different methods will give a reasonable estimate on the systematic effects.

In most cases the energies of the different methods are very compatible with each other. Sometimes the energies of the weight-shift-reweight and multi-state fit deviate in a statistically significant way and cannot be explained with a badly chosen fit range. In these cases it is not clear which of the two is closer to the true energy of that state.

3.4. Prony GEVM

A fundamentally different approach to the issue of thermal states is also possible: fitting earlier. The presence of excited states in the region of small t forces us to seek a plateau in a later region of the correlator when the excited states have already sufficiently decayed to be below the gauge noise level. The larger time slice we use, the more pronounced the thermal states get. If we were able to attenuate the excited states then we would have a plateau much earlier, being able to ignore the thermal states in that region. The Prony GEVM, *PGEVM* (or Hankel method in the plots) allows to separate away part of the excited states. This section will introduce the method and its application to the data at hand. More details about the method itself can be found in Reference [20], where we have published the details.

The GEVP resolves as many energies as there are operators put in. The tower of states that couple to operators is infinitely high, therefore even with a large GEVP there will still be excited states spoiling the first few time slices. In most cases they are sufficiently large and decay rather quickly, in other cases the excited states have not decayed before the noise towards the end of the temporal extent sets in. The principal correlators contain the state of interest as well as excited states that the GEVP could not resolve. The energy gap in the contributions in each principal correlator allows to treat the excited states as one or few effective states above the relative ground state. By constructing a GEVP from each principal correlator these excited states can in theory be resolved and separated from the actual signal.

For the construction we need the prescription for a Hankel matrix H of size $n \times n$ parametrized by a vector with $2n-1$ elements $\boldsymbol{x} = (x_0, x_1, \dots, x_{2n-2})$:

$$H(\boldsymbol{x}, n) = \begin{pmatrix} x_0 & x_1 & x_2 & \cdots & x_{n-1} \\ x_1 & x_2 & x_3 & \cdots & x_n \\ x_2 & x_3 & x_4 & \cdots & x_{n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n-1} & x_n & x_{n+1} & \cdots & x_{2n-2} \end{pmatrix} .$$

If we choose n smaller, then not all elements of the vector are used to build the matrix.

We define two vectors from the correlation function. The first is just the correlator starting from a reference time t_0, the second starts shifted by δt and therefore is shorter, only supporting a Hankel matrix up to $n < (T/2 - t_0 - \delta t)/2$. These vectors are

$$\begin{aligned} \boldsymbol{x} &= (C(t_0), C(t_0+1), \dots, C(T/2)) , \\ \boldsymbol{y} &= (C(t_0+\delta t), C(t_0+\delta t+1), \dots, C(T/2)) . \end{aligned}$$

Using the two vectors one can define a second GEVP with eigenvectors v and eigenvalues λ as

$$H(\boldsymbol{y}, n)\, v(t_0, \delta t) = H(\boldsymbol{x}, n)\, \lambda(t_0, \delta t)\, v(t_0, \delta t) .$$

To make sense out of the eigenvalues we need to assume the correlation function to be free of a back-propagating part and just a tower of exponentials,

$$C(t) = \sum_n a_n \exp(-E_i t) \,.$$

The eigenvalues that we obtain correspond to a single such exponential stripped of its amplitude,

$$\lambda_i(t_0, \delta t) = \exp(-E_i \delta t) \,.$$

Evaluating this for every possible time $t = t_0 + \delta t$ can be done in two different ways:

1. Keep t_0 fixed and let δt vary. The resulting eigenvalue functions will decay just as regular principal correlators would do. We will call this the “fixed PGEVM”.
2. Vary t_0 and keep δt fixed. The resulting eigenvalue functions will be a constant plus noise. In this work it is called “running PGEVM”.

As will be seen shortly, both methods yield different quality results.

3.4.1. Running PGEVM sorting issues

The running PGEVM severely suffers from sorting problems whereas the fixed PGEVM only shows this as much as the GEVPs from correlator matrices. The reason could be that a small energy gap in the eigenvalue expression $\exp(-E_i \delta t)$ with δt fixed to 1 will lead to a bad eigenvalue separation. The fixed PGEVM will have eigenvalues that are suppressed by $\delta t \geq 5$ in the cases used in this work.

The correlator matrices that form the first GEVP can be of arbitrary size. Eigenvalue sorting methods needs to work independent of the involved number. With both variants of the PGEVM one can freely choose the size n of these matrices, also controlling the number of eigenvalues that come out. From the data we have seen that only $n = 2$ gives sensible results, larger n make the signals worse and not better. For two eigenvalues a fourth sorting option is possible, which works as follows. From the other energy extraction methods as well as the non-interacting energies we expect the lower eigenvalue from the running PGEVM to lie in a certain energy corridor. The width needs to be tried out, for the middle one just uses a measurement. For the original data, only eigenvalues that lie within this

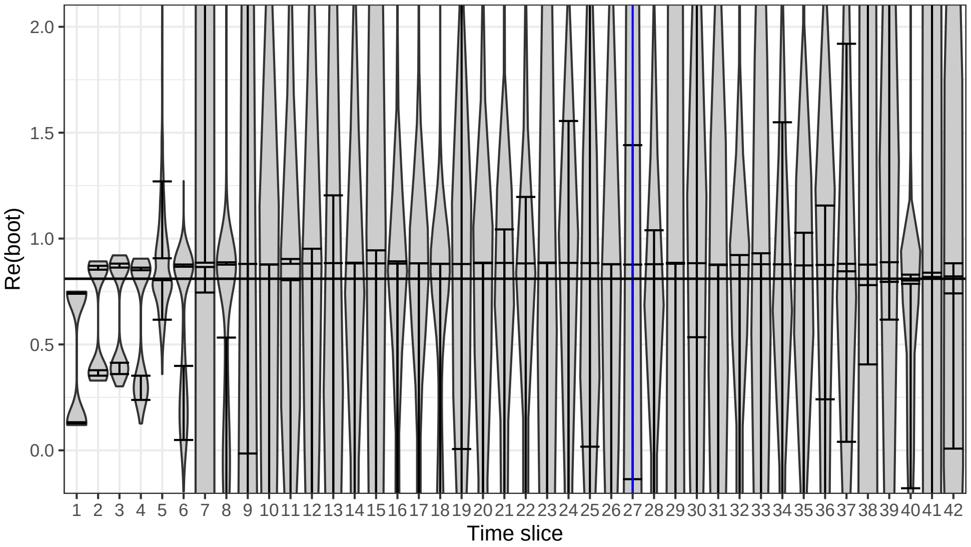


Figure 3.9.: Eigenspectrum of the running PGEVM per time slice for an examplaric case. cA2.09.48 ensemble, 2π channel, $P^2 = 0$, $\Gamma = A_1^+$, first principal correlator, weight-shift-reweight applied, bootstrap resampling. Details in the main text.

range are kept. This way the eigenvalue that corresponds to the signal of interest is chosen. Then for the samples the eigenvalue that is closer to the selected central one is picked. This way outliers in the eigenvalue spectrum are discarded. Still this method can fail when the central value lies far off the expected ground state.

Figure 3.9 shows the resampling distribution of both eigenvalues per time slice. Ideally these distributions should have two humps and should be separable. For the first four time slices in the given example this is quite easily possible. Also they lie close to the value determined with non-PGEVM as indicated by the horizontal black line. Using the corridor approach will lead to well separated eigenvalues. The error bars indicate the confidence interval for the two eigenvalues that the distribution has been divided into using the *range* sorting method.

For the later time slices the situation is not as clear. There the distribution ceases to have two humps and instead becomes one very broach one. Trying to classify that into eigenvalues seems to become very hard. Through this method the eigenvalue resampling distribution for the selected eigenvalues can be very wide, to the point where it becomes unusable. The vertical blue lines denote the time from which onward this principal correlator mixes with the next one in the first GEVP. Large noise beyond these lines therefore has nothing to do with the PGEVM and was introduced earlier on.

Mean and standard deviation become meaningless due to the very long tails of the distribution. Using an outlier resistant statistical treatment makes these time slices workable again. Instead of taking the standard deviation as a measure of uncertainty for the two eigenvalues, we use the half the distance between the 16 % and 84 % quantiles of the resampling distribution. The central value can also be an outlier, therefore we do replace the central value with the middle of the two quantiles. This will make the error estimate symmetric. Using this new central value and error estimate, parametric bootstrap could be used to generate the desired number of samples again. Correlation would be lost, however.

The covariance, being the multivariate generalization of the (squared) standard deviation, is also impeded by long tails. We deal with this as follows. Values that lie a certain number (we used 5.0) error estimates away from the central values are just ignored in the computation. The elements of the covariance matrix are computed for a pair of time slices, and pairs are used if neither value lies outside the respective thresholds. This pairwise nature could potentially lead to a covariance matrix that has negative eigenvalues [44], this has not posed any trouble in actual applications. The such determined covariance matrix is used in parametric bootstrapping to generate the desired number of samples without extreme outliers. The analysis continues as usual after this.

In Figure 3.10 the values and error estimates for the usual central value and standard deviation (black points and red bars) as well as the quantile method (green bars) are shown. For the first few time slices both methods coincide, the lack of outliers means that no tails are clipped. Then for the later time slices it can be seen that the quantile method is much more robust. For $t = 19$ we can also see that the original data (black point) is outside the confidence interval determined by the quantiles (green bar). The two quantiles however are just at the plateau, showing that it is an improvement to also resample the central value from the clipped bootstrap distribution. And at $t = 24$ one can see that the central value even lies outside of the plotting region, whereas the interval set by the quantiles agrees with the plateau.

The jackknife distribution is more narrow than the bootstrap distribution, therefore the sorting problem in the running PGEVM is much less severe. We have found that the signal quality of the jackknife data is just slightly worse than the cut and resampled bootstrap data.

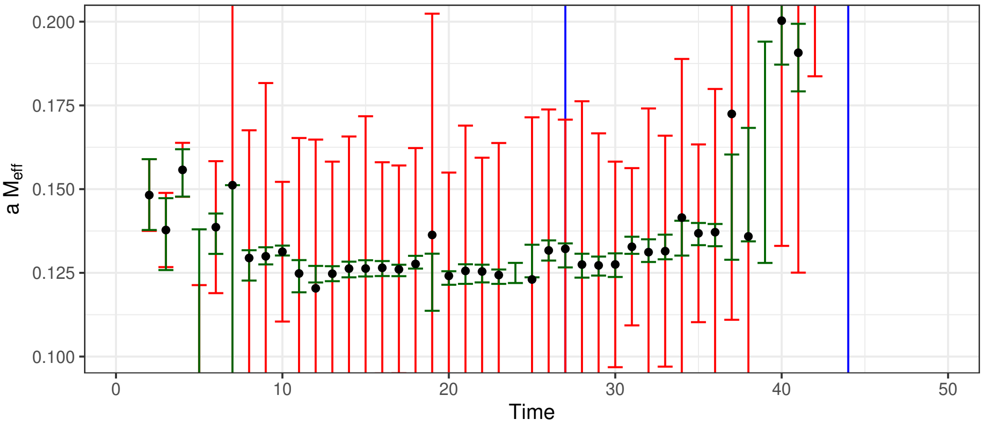


Figure 3.10.: Resampled ground state eigenvalue from the running PGEVM per time slice for an examplaric case. cA2.09.48 ensemble, 2π channel, $P^2 = 0$, $\Gamma = A_1^+$, first principal correlator, weight-shift-reweight applied, bootstrap resampling. Details in the main text.

3.4.2. Examples for the PGEVM

For illustration of both the running and fixed PGEVM we will take a look at the cA2.30.48 ensemble, 2π channel, $d^2 = 0$ and A_1^+ irrep, first principal correlator. This state suffers from thermal pollution such that there is a significant difference between weight-shift-reweight and no thermal state treatment. This state also has contamination with excited states, just like every other state used in this work. This way we can see the effects of both thermal and excited state treatment in interaction.

Figure 3.11 has six effective masses, the three methods (regular, fixed and running PGEVM) in combination with and without weight-shift-reweight. This will be the first plot in a series of three, this first one shows the overall behavior of the effective masses. There are two striking observations: The regular (non-PGEVM) method shown in red has significant excited states. It cannot be really seen at this zoom level, but there cannot be any plateau before $t = 10$. Then secondly the thermal pollution gives a downward trend towards large time separations if not treated (red and green circles). The difference between regular and fixed PGEVM is that for the regular method (red circles) we take the "solve" effective mass which takes the back propagating part into account but no thermal states, for the fixed PGEVM (green circles) we take the "log" effective mass which neither includes

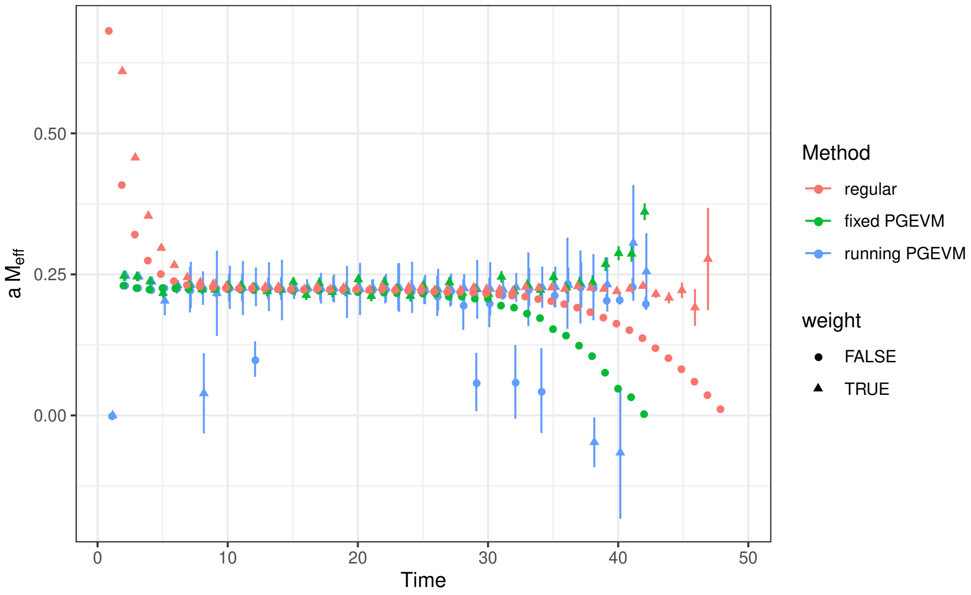


Figure 3.11.: Effective masses for different thermal and excited state treatmens. cA2.30.48 ensemble, 2π channel, $d^2 = 0$ and A_1^+ irrep, first principal correlator, weight-shift-reweight applied, bootstrap resampling.

thermal nor back propagating parts. In either case one cannot expect plateau in these regions.

Figure 3.12 then visualizes the same data, but zoomed in to see the plateaus better. Also the two different treatments of thermal states are separated into two facets of the plot. In both cases it can be seen that the fixed PGEVM suppresses the excited states during early times compared to the regular (no PGEVM) treatment. We also see that the running PGEVM seems to have a plateau almost the full time range. The uncertainties are rather large, however. The general observation is that although the running PGEVM is theoretically promising, it fails to deliver energies that are more precise than with the regular or fixed PGEVM. Therefore in this work the running PGEVM only serves as a cross check for the other methods.

Figure 3.13 has the regular and fixed PGEVM only, shown further zoomed in to see their qualities. In both thermal treatment variants it can be seen that the regular method has much stronger excited states than the running PGEVM. Without weight-shift-reweight (upper facet) the uncertainty is much smaller and the effect of the thermal pollution in the form of a downwards trend is clearly visible. As

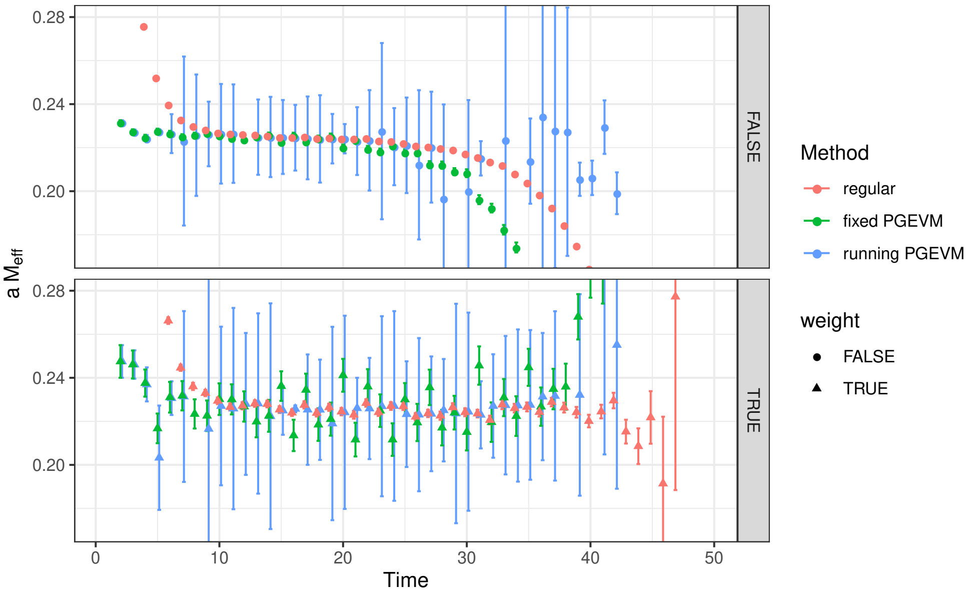


Figure 3.12.: Zoomed version of Figure 3.11. For better readability the plain and weighted correlators (`TRUE` and `FALSE`) have been separated into two panels in addition to the different shape.

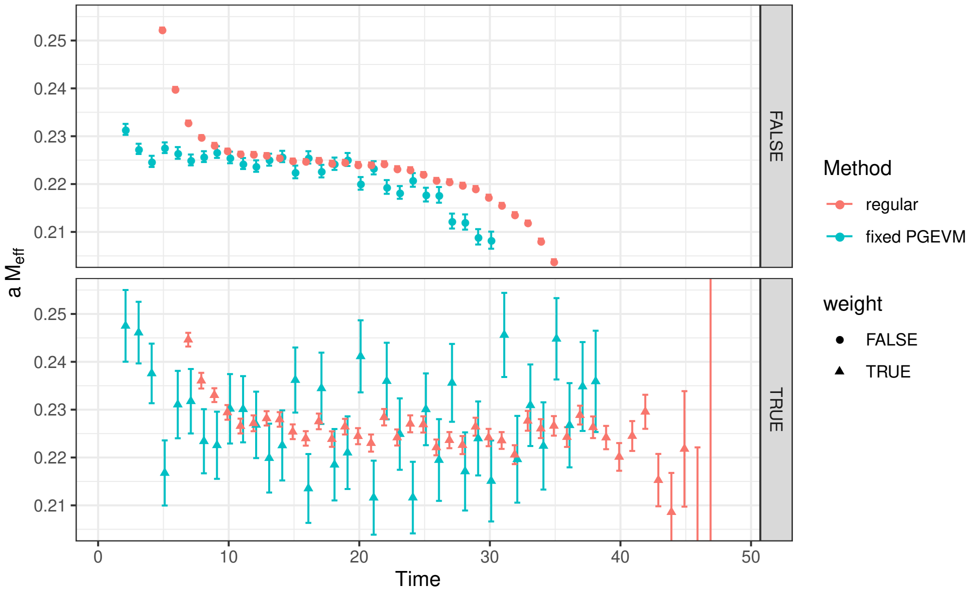


Figure 3.13.: Further zoomed version of Figure 3.12 without the *running PGEVM*.

the PGEVM does not take the back propagating part into account and only the “log” effective mass can be used for that model on the early time slices, it is not surprising that this method has an even stronger downwards trend compared to the regular method. However, the attenuation of excited states in the beginning lets one find a plateau at early time slices such that an energy can be extracted. The regular method can also be fitted with the multi-state model, there just is not an effective mass available for this.

In the lower facet one can see the same for correlators where weight-shift-reweight has been applied. The general noise level is much larger as the shift takes a numerically sensitive derivative of the correlator. The downwards trend is gone from both methods, the removal of the leading thermal pollution apparently was sufficient to make all further pollution drown in the gauge noise. For the regular method one can find a plateau and fit it with the corresponding “weighted” correlator model. For the fixed PGEVM one can find a plateau which starts a bit earlier. The noise level seems to be much larger in the data, yet the determined energy has roughly the same precision as with the regular method.

Fit ranges have been chosen for the various methods. The resulting fitted energies are compared in Figure 3.14. First one can see that the resampling method

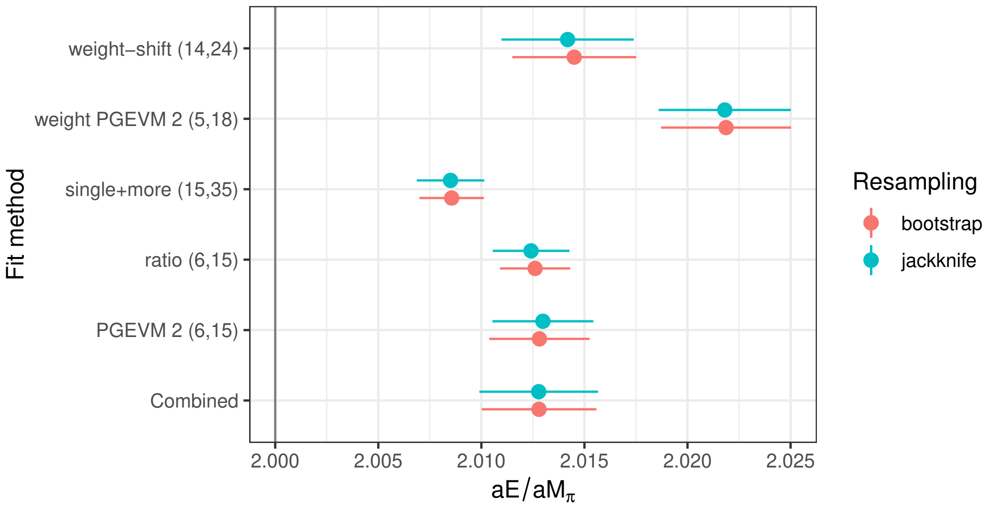


Figure 3.14.: Determined energies from the correlators shown in Figure 3.11. The fit method on the left axis also includes the chosen fit range in parentheses.

(bootstrap or jackknife) does not have a significant influence on the fit result. This means that jackknife is also sufficient for the error estimation during energy determination. Then one can see that the various methods have some tension between them, yet they are all compatible with a common average value. The combined value is computed with the procedure detailed in Section 3.6.

3.5. Ratio method

Instead of fitting $C_{2\pi}(t)$ and $C_\pi(t)$ separately and deal with excited and thermal states, one can form the ratio $C_{2\pi}(t)/C_\pi(t)^2$ which has an energy dependence of $E_{2\pi} - 2M_\pi$. Fitting this correlator ratio directly gives the energy shift from the interaction between the two pions. Additionally the thermal and excited states that are present in both correlators may cancel each other to some extent, giving a cleaner signal. This method has been used in various projects [45, 46] in a way that also includes the derivative method [47] to get rid of constant contributions.

For this work the two-particle ratio R_2 is used in the same way that it has been used by Feng, Jansen, and Renner [45]. For the three-particle ratio R_3 the ratio is taken in two steps; first a ratio, then the derivative, and another ratio. The forms used in this work are

$$R_2(t) = \frac{C_{2\pi}(t) - C_{2\pi}(t+1)}{C_\pi(t)^2 - C_\pi(t+1)^2} \tag{3.1}$$

$$R_3(t) = \frac{C_{3\pi}(t)/C_\pi(t) - C_{3\pi}(t+1)/C_\pi(t+1)}{C_\pi(t)^2 - C_\pi(t+1)^2}\,. \tag{3.2}$$

The three particle ratio has been chosen in this way to first remove the thermal states by a single pion with the ratio in the numerator. The numerator has a sinh-like shape such that dividing it by another sinh-like shape again gives a cosh-like shape. Other ratios have been tried out, this particular one seems to be the most effective one.

Figure 3.15 shows the effective masses for the three pion ground state $\boldsymbol{P}^2 = 0$, A_1^- irrep) on the cA2.60.32 ensemble, the one with the heaviest pion mass. Compared are the effective mass of just the correlator (plain), the one with weight-shift-reweight and the ratio R_3. The latter has been shifted upwards by the denominator energy to compare them with the other methods. One can see that the untreated correlator exhibits massive thermal states, no plateau can be identified. The effective mass of the weighted correlator looks much better and seems to show a plateau between $t_1 = 12$ and $t_2 = 20$; but there still is a downwards trend that likely arises from the second thermal state. The contributing thermal states in that state are shown in Figure 3.6 (Page 39). There one can see that the second state from the top still has a significant contribution, which has not been removed in the weight-shift-reweight procedure. The effective mass of the ratio features a much cleaner plateau.

In general we find that the ratio method allows the energy to be determined with less statistical uncertainty than other methods. In some cases, like cA2.09.48 at $\boldsymbol{P}^2 = 0$ and E^- irrep, the ratio method is the only one where a plateau can be identified. In rare cases we cannot find a convincing plateau in the ratio and rather omit this method in favor of the other methods.

One can also try to combine the ratio and the PGEVM assuming that the ratio behaves like a correlator. This usually fails because the ratio comes from below and then falls down again. The PGEVM tries to find the ground state, and with the ratio there are contributions with are not simply excited states. Almost no energy levels produce sensible results with the ratio PGEVM therefore this method was dropped completely.

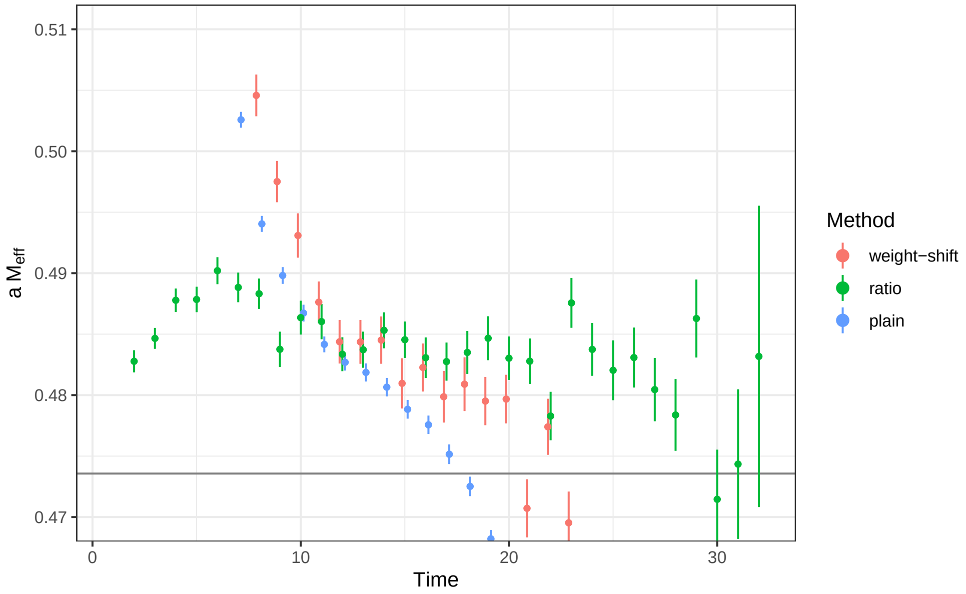


Figure 3.15.: Effective masses for the 3π ground state in the cA2.60.32 lattice. Shown are the effective mass for the plain correlator, the one with weight-shift-reweight and the one for the ratio (shifted upward by the denominator energy). The values are slightly dodged in the time direction to prevent overlaps. The horizontal line marks the non-interacting energy.

3.6. Combining results from different methods

As one can see in Figure 3.14, the different fit methods give results that are compatible with each other, yet show some tension. This tension is a good estimate for the systematic error in the fit range choice, for unresolved thermal states and unresolved excited states. Each of the statistical errors does not fully represent the uncertainty that the data really have. This becomes apparent when phase shift models are fitted to energies of a single fit method only. We will therefore use the information from all methods to compute one energy value with a combined statistical and systematic error that covers the whole uncertainty.

Let X_i be the bootstrap or jackknife resampling distributions of the energy values obtained by different fit methods, the indices i (and j) label the different methods. The individual samples are always labelled with $r \in [1, R]$. So one observation of one method would be x_{ir}. As usual in statistical settings distributions or statistical variables are denoted in capitals, single concrete values with lowercase letters.

First we computes weights for the different distributions using the variance:

$$w_i = \frac{\operatorname{var}(X_i)^{-1}}{\sum_j \operatorname{var}(X_j)^{-1}} \,.$$

As all distributions have the same number of elements, the jackknife scaling factor $(N-1)^2/N$ will drop out and can be ignored in this relation.

We further compute the weighted means for every sample r, we call the resulting quantity Y:

$$y_r = \sum_i w_i x_{ir} \,.$$

This quantity Y will have a smaller width than the individual distributions X_i. Correlation between configurations is preserved by this method such that energy shifts can still be computed in a meaningful way.

We compute the width of the resulting distribution of weighted means, which encodes the statistical error. For bootstrap resampling this is

$$a = \operatorname{sd}(Y) \,,$$

in case of jackknife resampling one need to include the scaling factor.

And additionally we compute the mean absolute differences of the individual central values compared to the weighted mean of the central values. This is an estimate for the systematic error between the various fit methods.

$$b = \mathrm{mean}(|\bar{X}_i - \bar{Y}|) \,.$$

We use the mean absolute error instead of the standard deviation as it is more robust against outliers [48, p. 41].

Using both the estimates for the statistical and systematic errors, a and b respectively, we can compute a scaling factor that should be applied. The general idea is that the statistical and systematic error are independent and should be added via the Pythagorean theorem. We want to keep the statistical error (1) and add the systematic error in terms of the statistical error (b^2/a^2). In total we obtain

$$s = \sqrt{1 + \frac{b^2}{a^2}} \,.$$

The distribution Y needs to be widened by that factor s via the prescription

$$\tilde{y}_r = (y_r - \bar{Y}) \cdot s + \bar{Y} \,.$$

The distribution Y will then contain both the statistical and systematic uncertainty as a simple normally distributed resampling distribution that is easy to work with.

For a simple test case with artificial data, the distributions X_1 and X_2 have been drawn from a normal distribution. In the case of “values: same” both central values are chosen as 1.0. In case of “values: different” the central values are 1.0 and 2.0. The standard deviation in “errors: same” is chosen to be 0.1 for both. In case of “errors: different” they are 0.1 and 0.3. Figure 3.16 shows the four possible combination of same/different values and errors for the distributions X_1, X_2, their weighted average and the weighted and rescaled distributions.

The four facets show distinct cases, which demonstrate the method in all possible combinations.

Bottom right The values and errors of both quantities are the same. We end up with a quantity that has a smaller overall error because we are able to use two independent measurements to yield the same result.

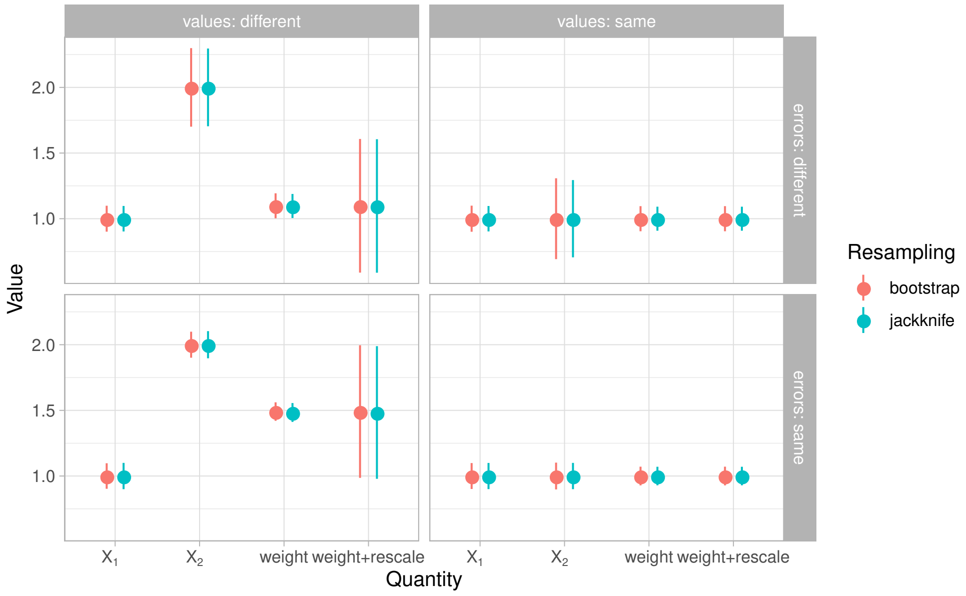


Figure 3.16.: Test example for combination of different resampling distributions using artificial data.

Top right Now the errors are different. The larger uncertainty of the second sample does not dilute the quality of the first sample.

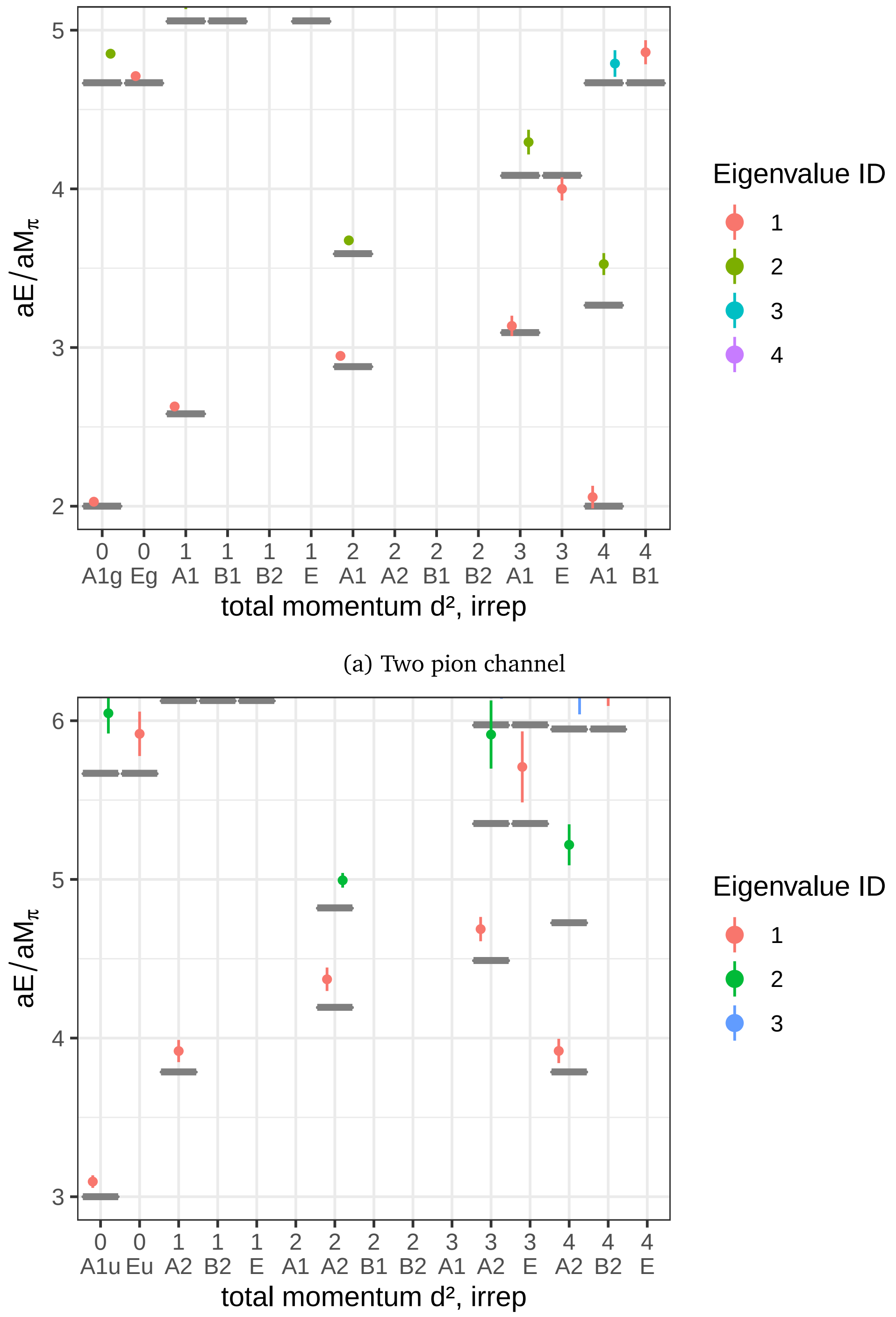
Top left The values are different now, but one sample is more precise than the other. The average value gets nudged a bit, but stays with the more precise value. However, the systematic error is now included and covers the other point in a reasonable way.

Bottom left The statistical errors seem to be rather low, but the systematic is very large. The resulting value is in the middle and the error covers both observation.

3.7. Results

The results of this chapter are the two-pion and three-pion spectra measured on all three ensembles. The spectra in each channel are displayed in the Figures 3.17, 3.18, 3.19. Only the interesting region until slightly above the respective inelastic thresholds are shown. One can see that in the ensembles with larger $M_\pi L$ there are more states below threshold. All energy levels are made publicly available [49] such that an independent analysis of the data can be performed.

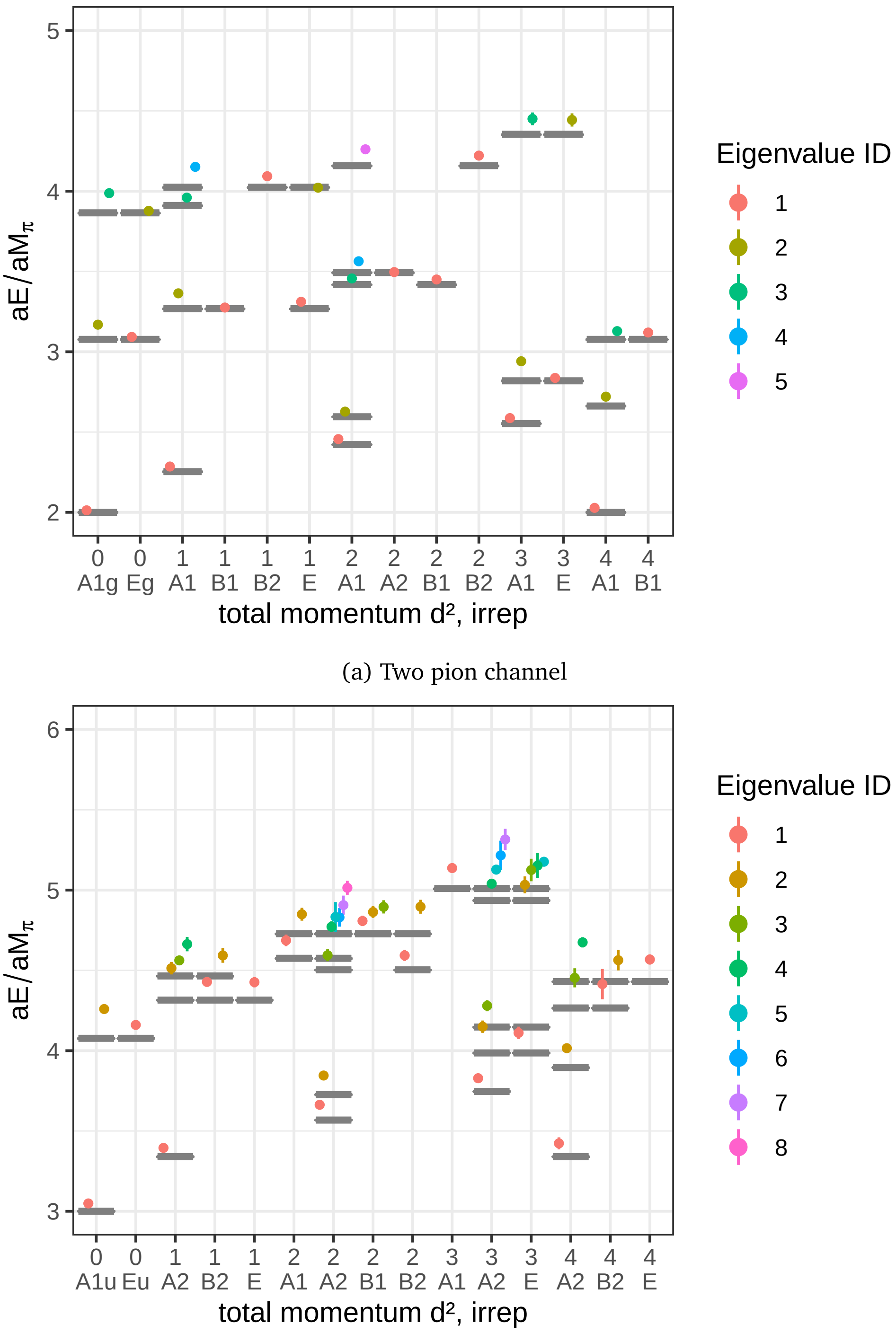
Most energy shifts (interacting compared to non-interacting) are statistically significant and repulsive. There are some cases in which the shift is compatible with zero. Whether the energy levels are consistent can only be seen by the χ^2 value of the fits to the spectrum which get introduced in the next two chapters.



(a) Two pion channel

(b) Three pion channel

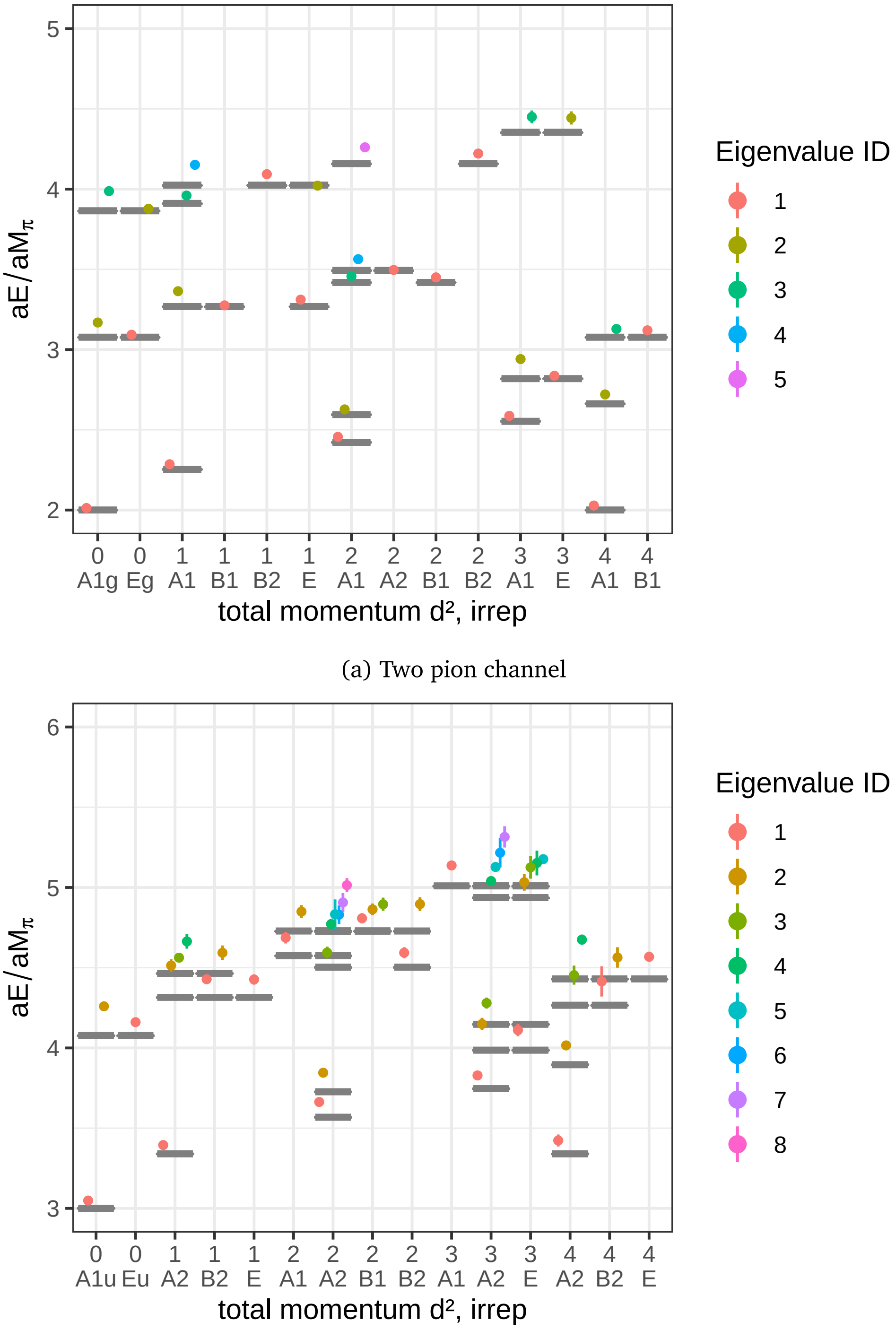
Figure 3.17.: Spectrum of cA2.09.48.



(a) Two pion channel

(b) Three pion channel

Figure 3.18.: Spectrum of cA2.30.48.



(a) Two pion channel

(b) Three pion channel

Figure 3.19.: Spectrum of cA2.60.32.

4. Two pion scattering

Scattering experiments have given many insights into the physics of elementary particles and their interactions. The LHC at CERN is known to many people outside of the physical community and performs such scattering experiments. The accelerated particle beams are brought onto a collision course, a surrounding detector measures final state particles created in the interaction at the center. One does not use pion beams directly as they are unstable particles under the weak and electromagnetic force. Experiments with pions in nature are more indirect than in lattice QCD simulations.

We want to review how scattering experiments are analyzed and then approach this with a lattice simulation from the theoretical side. Lattice QCD creates predictions that can then be compared with the experiment.

Scattering with two pions on the lattice has been done many times before. As they are the simplest asymptotic states they are the entry point and test bed for any analysis method. Usually one starts in the maximal isospin channel because the signals are the cleanest due to the lack of quark-disconnected diagrams. For an overview on two-particle lattice scattering, see for instance the review by Briceno, Dudek, and Young [50].

There are already many investigations at $I = 0$ [17, 51–53] and $I = 1$ [54–57] with different lattice actions, number of flavors and pion masses. There exist also a lot of studies for $I = 2$, a few of them are in the following list in rough chronological order.

- Feng, Jansen, and Renner [45] determined the S-wave scattering length for $N_\mathrm{f} = 2$ Wilson twisted mass fermions down to $M_\pi = 270\,\mathrm{MeV}$.

- Dudek et al. [58] presented a trial of their S- and D-wave extraction method on $N_\mathrm{f} = 3$ Wilson clover fermions with several masses down to $M_\pi = 400\,\mathrm{MeV}$.

- Dudek, Edwards, and Thomas [34] then followed up with a S- and D-wave extraction using a large correlator matrix on $N_\mathrm{f} = 3$ Wilson clover fermions down to $M_\pi = 396\,\mathrm{MeV}$.

- Fu [52] extrapolated the S-wave scattering length at the physical point using $N_\mathrm{f} = 2 + 1$ staggered quarks down to $M_\pi = 240\,\mathrm{MeV}$.

- The ETMC [59] used $N_\mathrm{f} = 2 + 1 + 1$ Wilson twisted mass fermions, with three lattice spacings, multiple volumes and pion masses down to 250 MeV to produce the S-wave scattering length at the physical point with a detailed investigation of systematic effects.

- Mai et al. [60] performed a combined analysis of two pion scattering at all three isospin channels on $N_\mathrm{f} = 2$ lattices and obtain the scattering lengths as well as ρ and σ pole positions.

As we will later see in Chapter 5, every three pion scattering analysis needs results from the two pion subsystem. In order to obtain such data in a correlated way, usually the two pion scattering is also performed in the works that will be introduced in said chapter.

4.1. Partial wave expansion

In scattering experiments one usually measures count rates with a detector covering as much of the solid angle as possible. With the proper normalization this gives the differential cross section $\mathrm{d}\sigma / \mathrm{d}\Omega$. Total angular momentum l is a conserved quantity as we work with a spherically symmetric interaction, we therefore have no dependence on the polar angle ϕ. The asymptotic scattering states can be written via plane waves of momentum k in spherical coordinates [61, (18.27)],

$$f_k(\theta) = \sum_{l=0}^{\infty} (2l + 1) f_l P_l(\cos\theta) \,,$$

with scattering amplitudes f_l. These amplitudes can be expressed as a *phase shift* angle δ_l and are related to the scattering amplitudes via [61, (18.37)]

$$f_l = \frac{\exp(\mathrm{i}\delta_l) \sin(\delta_l)}{k} \,.$$

Additionally we have elastic scattering, therefore the different partial waves do not mix. This yields a simple form of the differential cross section [61, (18.37)]:

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\Omega} = \frac{1}{k^2} \sum_l (2l+1)^2 \sin(\delta_l)^2 P_l(\cos\theta)^2 \,.$$

All information from the cross section therefore can be also expressed with the phase shifts δ_l.

The scattering momentum dependence can be expanded in a power series, the *effective range expansion*. To first order this is given by

$$k \cot(\delta_0) = -\frac{1}{a_0} + \frac{1}{2} r k^2 \,,$$

where k is the relative scattering momentum, δ_0 the phase shift of angular momentum $l = 0$, a_0 the scattering length and r the effective range. As this model has two parameters, it will be called "ERE-2" in this work. Depending on the physical field the definition has a different sign for a_0.

As also stated in Reference [62], there is a zero in the scattering amplitude for $k = M_\pi/2$ called the "Adler zero". The effective range therefore does not describe this region. For the two-pion scattering this is not a limitation. However, the two-pion subsystem in three-particle scattering can become sub-threshold; a description which is valid in that region is needed. Taking the Adler zero into account with a two parameter model gives the "Adler-2" model:

$$\frac{k}{M_\pi} \cot(\delta_0) = \frac{\sqrt{s}}{s - 2z^2} M_\pi^2 \left(B_0 + \frac{B_1}{M_\pi^2} k^2 \right) .$$

We fix $z^2 = M_\pi^2$ as this is the LO ChPT prediction. Even when left as a free parameter, the resulting value is compatible with LO ChPT [63–65] (via [40]).

The parameters B_0 and B_1 are related to the scattering length and effective range via

$$M_\pi a_0 = \frac{2 - z^2}{B_0} \,, \quad M_\pi^2 a_0 r = \frac{2 + z^2}{2 - z^2} - \frac{2B_1}{B_0} \,.$$

One can also add the next order as $B_2 M_\pi^2 / k^2$ into the parentheses to make it the "Adler-3" model.

The scattering length and effective range can be expressed via chiral perturbation theory (χPT or ChPT). To leading order we have the predictions

$$M_\pi a_0 = \frac{M_\pi^2}{16\pi F_\pi^2}\,, \quad M_\pi^2 a_0 r = 3\,.$$

There are different normalizations for F_π, one where it has the physical value of around 91 MeV and another where there is an additional factor of $\sqrt{2}$ such that the physical value is around 129 MeV. The ETMC uses the latter convention.

As can be seen in previous lattice studies, the leading order ChPT works surprisingly well for this S-wave scattering phase shift at maximum isospin [45].

Only few data points are available in irreps that couple to the d-wave interaction. Therefore we use a zeroth order expansion in this case [40],

$$\frac{k^5}{M_\pi^5}\cot(\delta_2) = -\frac{1}{M_\pi^5 a_2}\,.$$

4.2. Lüscher method

Lattice QCD needs to operate in imaginary time to benefit from importance sampling. However, scattering in imaginary time cannot directly yield access to real time scattering quantities like the phase shift. An indirect approach is needed. The one used in this work has been set out by Lüscher [66] and is called the *Lüscher method*. Initially it gave the scattering length for two equal particles at rest and was later extended for non-equal particles, non-zero relative momenta and non-zero total momentum. Relatively new are extensions to more than two particles. For the two pion scattering in this work the formulation by Göckeler et al. [35] is used.

The alternative method for indirect scattering is the *HAL QCD* method which works with an effective QCD potential between the particles and computes scattering quantities from that.

All forms of the Lüscher method connect the infinite volume scattering quantities (like the phase shift or scattering length) to the finite size energies. The core is the quantization condition which for the two pion system takes the form given by

Göckeler et al. [35, Equation 54]:

$$\det(M_{lm,l'm'} - \delta_{ll'}\delta_{mm'}\cot(\delta_l)) = 0\,. \tag{4.1}$$

We investigate elastic scattering, therefore we will always have $l = l'$. The notation is to be understood such that the determinant is taken of a tensorial quantity with the indices $lm, l'm'$ where the second summand is a diagonal.

The matrix $M_{lm,l'm'}$ is a function of the relative back-to-back momenta of the two particles. It is expressed in the continuum angular momentum quantum numbers l and m which we do not have on the lattice. We rather have our data points in the lattice irreps Γ. Depending on the continuum angular momentum l and l' we will have a different *subduction* of the continuum scattering amplitude $M_{lm,l'm'}$ into the lattice one with the following subduction prescription [35, Equation 60] using the subduction coefficients from Equation (2.3):

$$M^{\Gamma}_{ln,l'n'} = \sum_{mm'} c^{\Gamma\alpha n *}_{lm} c^{\Gamma\alpha n'}_{l'm'} M_{lm,l'm'}\,.$$

Using this change of basis we obtain the applicable form of the quantization condition Göckeler et al. [35, Equation 63]:

$$\det\left(M^{\Gamma}_{ln,l'n'} - \delta_{ll'}\delta_{nn'}\cot(\delta_l)\right) = 0\,.$$

Again this is to be understood as the determinant of a tensor with the given index structure.

In the continuum the different partial waves are completely orthogonal to each other, there is no mixing between partial wave. The subduction of infinitely many continuum angular momenta l to the finite number of lattice irreps Γ leads to a mixture. The partial waves mix on the lattice and they need to be treated together in principle. Assuming that higher partial waves are sufficiently suppressed by the centrifugal barrier, one can just take a look at the leading partial wave for any given irrep. Table 2.2 tallies this mixing and allows us to investigate S-wave and D-wave by taking the appropriate irreps and ignoring higher waves. The quantization condition becomes the determinant of a scalar quantity and can therefore directly be inverted. For the ρ we have fixed it to $l = 1$, for two pions in S-wave we will fix it to $l = 0$. All other elements of the continuum scattering amplitude are then taken as zero.

A proper combined treatment of the different partial waves will need to solve the determinant equation with multiple data points for all phase shifts at the same time, usually using a fit to the energy levels.

The continuum scattering amplitudes are themselves expressed via linear combinations of the w_{js} given by Clebsch-Gordan coefficients C [35, Equation 39]:

$$M_{lm,l'm'} = \sum_{j=|l-l'|}^{l+l'} \sum_{s=-j}^{j} \sqrt{2j+1}(-1)^l \mathrm{i}^j \, C_{lm,js,l'm'} \, w_{js}(q^2) \,.$$

The Clebsch-Gordan coefficients are here expressed in the Wigner 3j-symbols as [35, Equation 35]

$$C_{lm,js,l'm'} = (-1)^{m'} \mathrm{i}^{l-j+l'} \sqrt{[2l+1][2j+1][2l'+1]} \begin{pmatrix} l & j & l' \\ m & s & -m' \end{pmatrix} \begin{pmatrix} l & j & l' \\ 0 & 0 & 0 \end{pmatrix} .$$

The w_{js} themselves consist of the Lüscher Z-function together with the Lorentz boost factor γ [35, Equation 40]:

$$w_{js}(q^2) = \frac{\gamma^{-1}}{\pi^{3/2}\sqrt{2j+1}} q^{-j-1} Z_{js}(1, q^2) \,.$$

This function can be computed as

$$Z_{js}(1, q^2) = \sum_{\boldsymbol{z} \in P_{\boldsymbol{d}}} \frac{|\boldsymbol{z}|^l \mathrm{Y}_{js}(\boldsymbol{z}/|\boldsymbol{z}|)}{\boldsymbol{z}^2 - q^2} \quad \text{with} \quad P_{\boldsymbol{d}} = \left\{ \boldsymbol{\gamma}^{-1} \left[\boldsymbol{n} - \frac{1}{2}\boldsymbol{d} \right], \boldsymbol{n} \in \mathbf{Z}^3 \right\} ,$$

where $\boldsymbol{d}$ is the integer total momentum and $\boldsymbol{\gamma}$ is a matrix applying the Lorentz boost factor to the boosted dimension only [35, Equations 33f]. The grid $P_{\boldsymbol{d}}$ has points for all the non-interacting relative momenta, $Z(1, q^2)$ therefore has poles there. This function has multiple numeric implementations, we use the R implementation *rzeta* [67].

4.2.1. Singularities in the Zeta function

Lüscher's Z-function depends on the relative momentum $\boldsymbol{q}$ and the Lorentz boost factor γ. It has poles exactly on non-interacting kinematic configurations where two particles have lattice momenta. Sometimes the energy of a certain state is somewhat compatible with no interaction. In these cases the samples (from

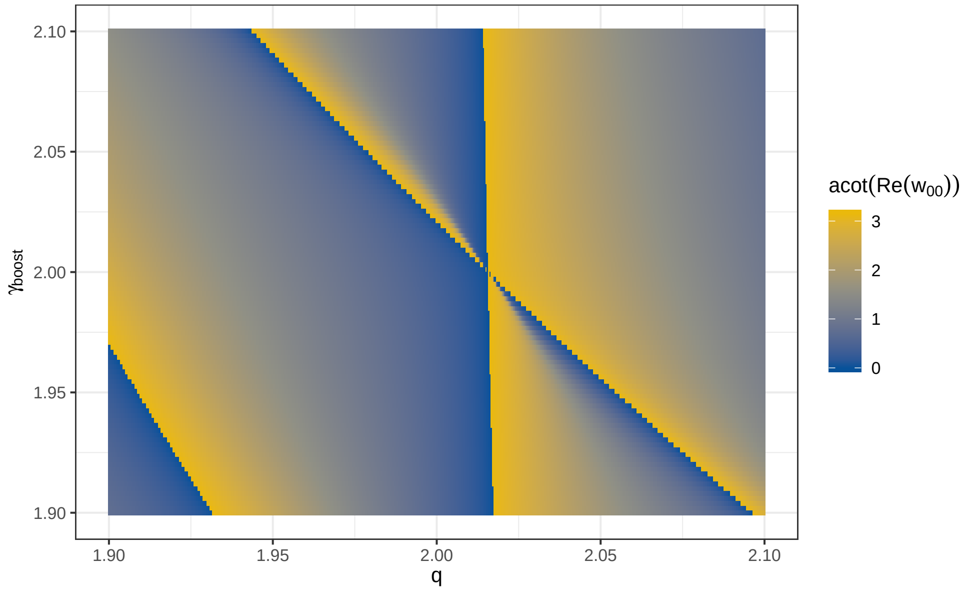


Figure 4.1.: Phase shift computed from w_{00} as a function of relative momentum q and Lorentz boost factor γ. The colored edges denote jumps from $\delta = 0$ to $\delta = \pi$ and accur at the singularities of the $Z(1, q^2)$ which also depend on γ.

bootstrap or jackknife) cross the non-interacting energy level. This leads to a problem in the Z-function as the associated phase shift usually jumps from $\delta_l = 0$ to $\delta_l = \pi$. This effect is illustrated in Figure 4.1. The resampling distribution of the phase shift will not be normally distributed but rather have half a peak at $\delta_l = 0$ and another half peak at $\delta_l = \pi$. Taking the standard deviation of this distribution will give a much overestimated error.

The crossing of singularities means that the samples are compatible with both a repulsive and an attractive state. Various ideas on the treatment were formed, in the end the conclusion has been that these problematic energy levels should simply be excluded. A promising idea seemed to check whether only a few of the samples cross the singularity. In these cases the distribution could be clipped at the tails and instead of using a standard deviation one would instead use quantiles which are insensitive to removal of the tails. The usual standard deviation corresponds to the 16 % and 84 % quantiles, clipping a few percent of the points on each end therefore does not hurt. The problems arise in further transformations which might not be strictly monotonic and therefore would depend on the tails.

The jackknife distribution is much less wide than the bootstrap distribution. A common way to work around the singularity crossing of the bootstrap distribution is to use jackknife as a resampling method instead. As jackknife is only a linear approximation to the bootstrap we expect that it becomes less accurate for large non-linearities. The number of points in the phase shift are much less decimated when using jackknife and allow for a more sensible extrapolations in the following steps.

4.3. Fitting the phase shift

We have seen how the phase shift is related to the energy levels via the Lüscher formalism. As a first step one can just focus on S-wave and has a direct correspondence without a need for a scattering matrix model. Also we have seen how the phase shift can be modeled. Now there are two ways to extract the model parameters from the energy levels determined in Chapter 3:

1. In a two-step process one first computes the S-wave phase shift δ_0 for every energy level. There will be strong correlation between the relative scattering momentum aq and the phase shift δ_0, which needs to be preserved via resampling.

   Then one fits the phase shift model to the points (aq, δ_0) while taking into account the covariance between independent and dependent variable as well as among the points. Fitting with uncertainties on the independent variable is done by promoting the values to fit parameters and adding priors to the χ^2 expression. With N points and n model parameters one needs to work with $N + n$ fit parameters.

2. Combine the Z-function directly with the fitting and compute δ_0 dynamically during the iterations of the fit. This way the fit parameters are solely the ones of the model. Also correlation between scattering momentum aq and phase shift δ_0 is taken into account automatically. There is no need for a covariance between them as the dependent variable is the energy and the independent are just nominal labels for the various energy levels (total momentum, irrep, correlator id).

The second method seems to be rather popular as it has less fit parameters to work with. The only caveat seems to be that the rather expensive Z-function needs to be evaluated for every iteration of the fit and not just once per energy level and bootstrap sample. Additional benefit is that one does not compute a resampling distribution of the phase shift and is safe from the singularity crossing. The samples of the energy levels can still be part repulsive and part attractive, but the distribution as a whole does not enter the fit via covariance as it does in the first method.

The two-step process has been used in Reference [30]. The same code was applied to the data of this thesis to cross-check the implementation of the second method that Fernando Romero López has used to analyze the same data.

A direct comparison using both methods with the same method and the same data shows that there are discrepancies in the resulting fit parameters. After exclusion of common errors and cross-checking intermediate results, this is interpreted as a systematic problem of the first method. By evaluating the Z-function only on the bootstrap samples, the fit model does not know anything about the Z-function. It is possible that aq, which become fit parameters, vary such that $\delta_0(aq)$ does not match the previously computed value of the phase shift. This would yield phase shift model parameters that could not actually be fulfilled with the given energy levels.

4.4. Results

The energies of the A_1 irreps contributing to the s-wave phase shift are fitted per ensemble using both ERE and Adler models. The ERE-2 model produces results which are compatible with the Adler-2 result. As the latter model better describes the data, we show the results for the Adler-2 and Adler-3 models in Table 4.1. As the two-pion interaction is also heavily used as part of the three-pion interaction, more results will be found in the combined fits that are presented in Section 5.2. For the physical point ensemble the results of the fit with the Adler-2 model are visualized in Figure 4.2, together with fits to experimental data [65, 68] as a comparison.

Figure 4.3 shows the fit to the cA2.30.48 ensemble and Figure 4.4 shows the fit to the cA2.60.32 ensemble. As these feature unphysical pion masses, no comparison to experimental data is possible. Deviations from the leading order ChPT prediction

Ensemble	$1/B_0$	B_1	B_2	χ^2/dof
cA2.09.48	$-0.0477(90)$	$-1.4(1.2)$	—	$11.08/(10-2)$
cA2.30.48	$-0.160(23)$	$-1.8(6)$	—	$15.30/(16-2)$
cA2.60.32	$-0.2090(54)$	$-2.3(3)$	—	$19.06/(16-2)$
cA2.60.32	$-0.2110(57)$	$-3.1(6)$	$0.4(2)$	$15.96/(16-3)$

Table 4.1.: Results for s-wave fits using the Adler-2 and Adler-3 models with fixed $z_2^2 = M_\pi^2$. Table taken from Reference [40].

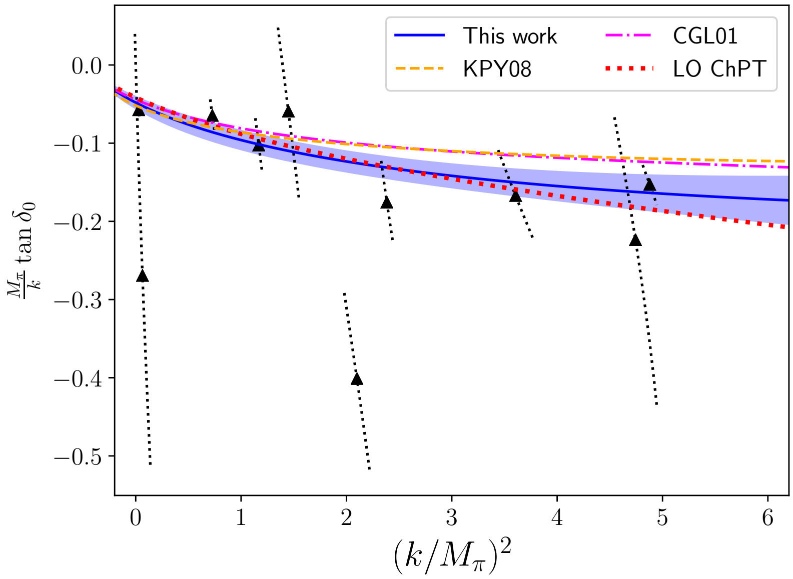


Figure 4.2.: s-wave phase shift fitted with the Adler-2 model on the physical point ensemble cA2.09.48. Our results are compared to the fits to experimental data by Kaminski, Pelaez, and Yndurain [65] and Colangelo, Gasser, and Leutwyler [68]. Figure taken from Reference [40].

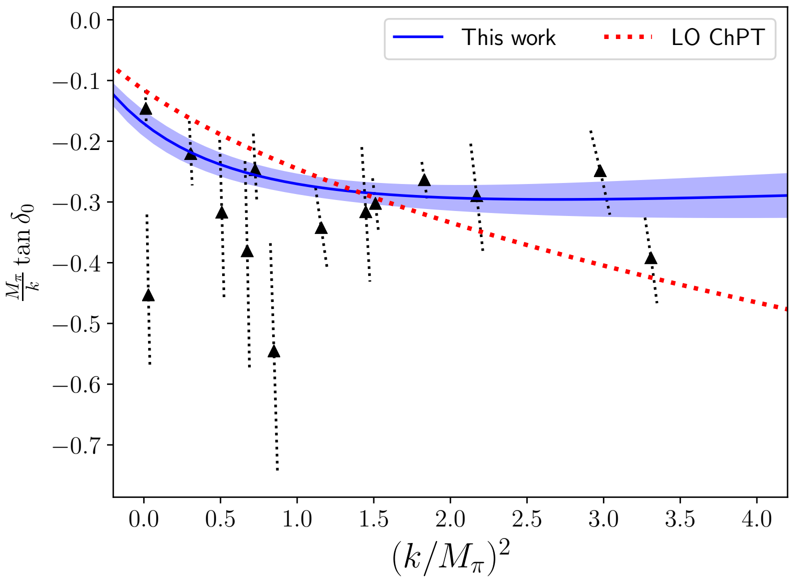


Figure 4.3.: s-wave phase shift fitted with the Adler-2 model on cA2.30.48. Figure taken from Reference [40].

become more pronounced with heavier pion masses, which of course is not a surprise as the pion mass is the expansion parameter.

In Table 4.2 the results of the d-wave phase shift model fitted to the energies in the irreps contributing are listed. A visualization of the results is presented in Figure 4.5, together with a fit to experimental data [65]. Their result is a repulsive d-wave interaction in the energy region which we also have in our spectrum. Our result lies lower and also suggests a repulsive interaction with mild statistical significance. The gap between our and their result is slightly more than one standard deviation, we therefore deem them as compatible. The range of sampled energies does not extend low enough towards the threshold that we could make any statement about the zero crossing in the phenomenological phase shift fits.

Figure 4.6 and 4.7 show our phase shift fits to the two heavier ensembles. Again, as these have unphysical pion masses, there is no comparison to physical results.

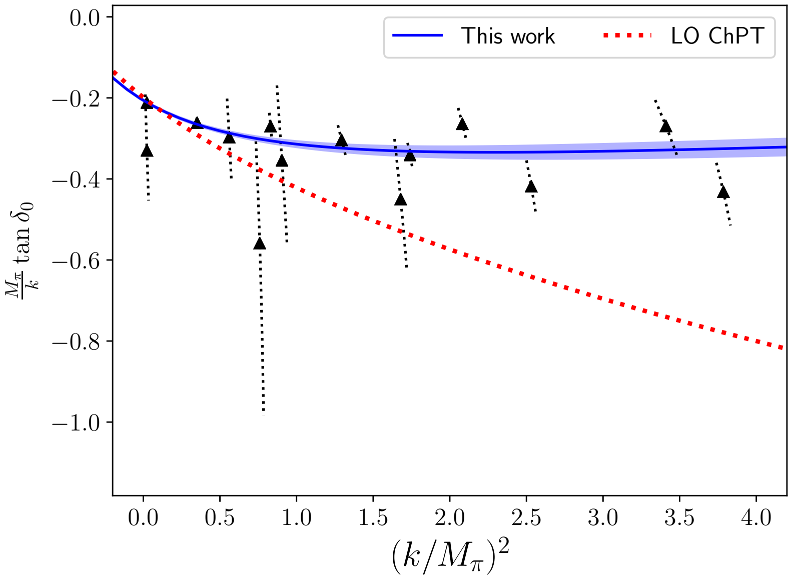


Figure 4.4.: s-wave phase shift fitted with the Adler-2 model on cA2.60.32. Figure taken from Reference [40].

Ensemble	$M_\pi^5 a_2$	χ^2/dof	E_{CM} range
cA2.09.48	0.0005(3)	$7.33/(4-1)$	$[4.0M_\pi, 6.3M_\pi]$
cA2.30.48	0.007(2)	$16.89/(10-1)$	$[2.8M_\pi, 4.2M_\pi]$
cA2.60.32	0.0037(8)	$15.03/(12-1)$	$[3.2M_\pi, 4.4M_\pi]$

Table 4.2.: Results for d-wave fits using a constant model. Table taken from Reference [40].

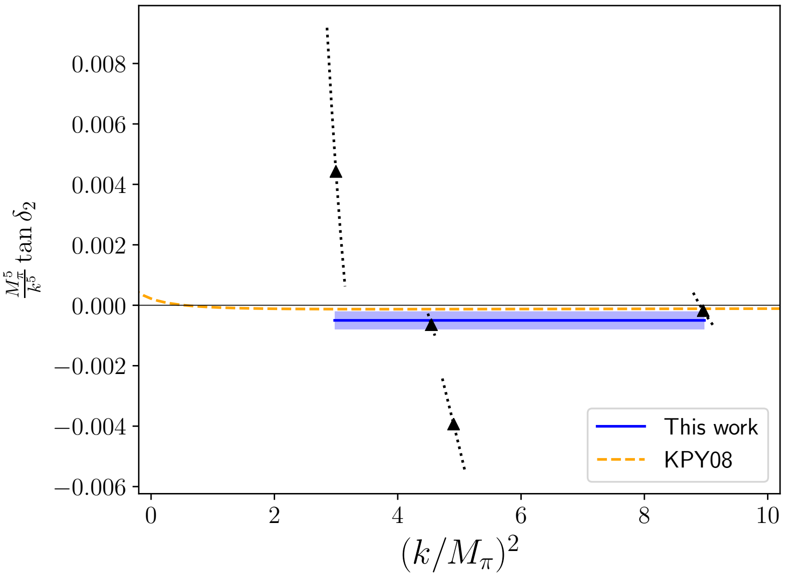


Figure 4.5.: d-wave phase shift fitted with a constant model on the physical point ensemble (cA2.09.48). Our results are compared to the fits to experimental data by [65]. Figure taken from Reference [40].

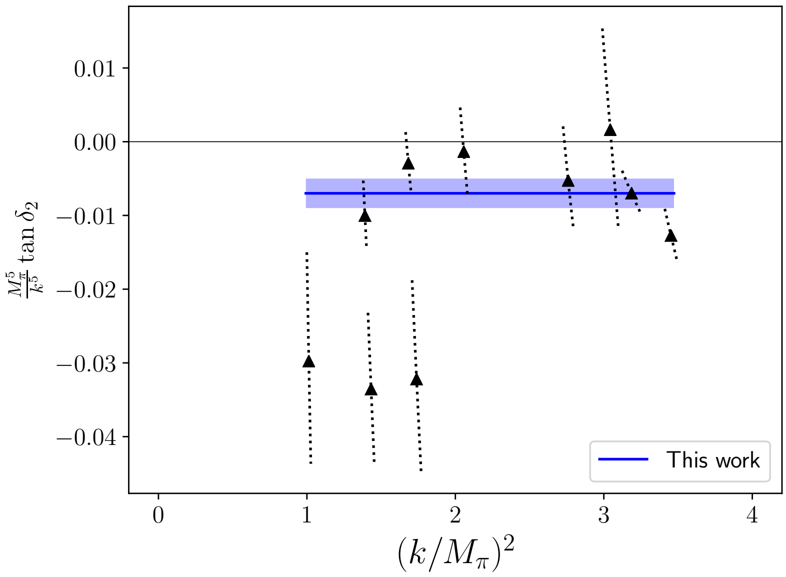


Figure 4.6.: d-wave phase shift fitted with a constant model on cA2.30.48. Figure taken from Reference [40].

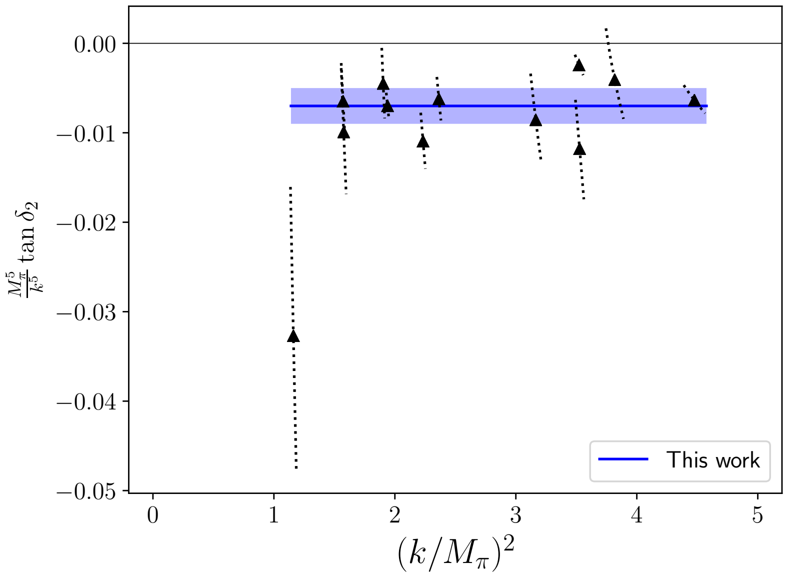


Figure 4.7.: d-wave phase shift fitted with a constant model on cA2.60.32. Figure taken from Reference [40].

5. Three pion scattering

Scattering with two particles is well established. Symmetric two-particle interactions of spin-less particles are reduced to a one particle problem by boosting into the center of mass system and treating it as a single particle in a possibly complicated potential. Counting the degrees of freedom also shows that a two-particle system is rather simple. We have 6 kinematic degrees of freedom (position, momentum) for a single particle. Two particles have 12 degrees of freedom, but there are 10 conserved quantities (center of mass movement, total angular momentum, energy), bringing this down to only 2 degrees of freedom, basically the two axes of an ellipsis or the independent Mandelstam variables s and t.

Adding a third particle makes this extremely more complicated as no transformation can reduce it to a one-particle system. The degrees of freedom are $3 \cdot 6 - 10 = 8$ and more than twice as many as before. The kinematic machinery needed to analyze such a scattering process is much more complicated and has only been built up recently.

The motivations for studying multi-particle scattering are manifold. As pion masses closer or at the physical value are possible, the relevant thresholds become lower as well. Without a proper description of the physics beyond the $4M_\pi$ and similar thresholds one cannot reliably extract scattering information at these energies. Also there are resonances that decay into three or even more particles, like the Roper resonance [1] (via [62]) and many of the X, Y, Z resonances [62] or the ω-meson [2]. The first steps into a new formalism are made with physically easy systems. At the current state the investigation of three pions with maximum isospin provides the simplest case in QCD to apply the formalism and advance the field.

A recent review of the three particle scattering formalism is given by Hansen and Sharpe [69]. There are three approaches, a relativistic field theory approach (RFT), a non-relativistic effective field theory approach (NREFT) and a finite volume unitarity approach (FVU). In this work the RFT approach will be used as it is the only one extended for higher partial waves in the two particle subsystem at this time.

The following is a selection of recent works that made progress in the three pion scattering and lay the foundation of this work.

- Beane et al. [70] computed the ground-state energies ($P^2 = 0$) for up to five π^+ using LQCD. They extracted the two-particle scattering length and also find a repulsive three pion interaction. Used are MILC lattices with $M_\pi \geq 291\,\text{MeV}$ [71] and domain wall valence fermions.

- Detmold et al. [72] extended their previous work [70] to include twelve π^+. Their finding of a repulsive three pion interaction is strengthened.

- Romero-López, Rusetsky, and Urbach [73] apply the threshold expansion in complex ϕ^4 theory.

- Mai and Doring [74] used experimentally available data for the 2π system and previous LQCD energy determinations of the three pion ground state [70, 72] to extract a three body coupling via the FVU approach.

- Hörz and Hanlon [32] were the first to provide energy levels for the three pion system at maximum isospin for all irreps $P^2 \leq 3$. The two pion system is also given until $P^2 \leq 4$. The energies are extracted on the $N_\text{f} = 2+1$ lattice with $M_\pi = 200\,\text{MeV}$ from the CLS effort.

- Blanton, Romero-López, and Sharpe [75] extended the RFT formalism to also include D-wave states for the dimer.

- Blanton, Romero-López, and Sharpe [62] analyzed the existing three pion energies [32] using the extended RFT formalism [75]. In their analysis they were able to obtain a significant non-zero contribution of the s-wave states of the dimer, the d-wave contributions were compatible with zero.

- Mai et al. [76] analyzed the existing data [32] using the FVU approach.

- Hansen, Romero-López, and Sharpe [77] have extended the RFT formalism to lower-than-maximum isospin channels.

- Beane et al. [78] have computed the ground-state energies of charged and mesons on two different lattices, once with up to 12 pions, and again with up to three nucleons. This way the electromagnetic effects could be quantified in such a many-particle environment.

- Culver et al. [79] compute three pion $I = 3$ energies on two ensembles (220 MeV and 315 MeV) using three elongations each. They show a generalization of the FVU approach to elongated boxes and use this to predict an expected spectrum using LECs from other sources. These are compared to the energy levels determined from the lattice scattering.

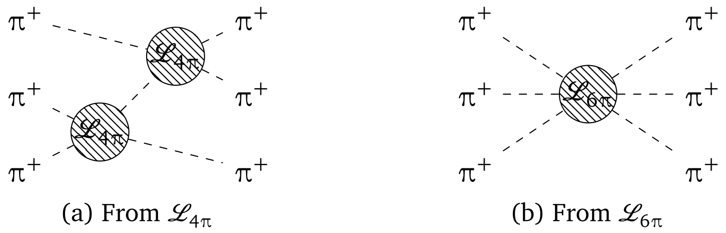
In this work we present an extension to multiple twisted mass ensembles including physical pion mass and an attempt to resolve the D-wave in the dimers. The lattices only have $N_\mathrm{f} = 2$ and also all share the same lattice spacing, making only an extrapolation (interpolation, actually) in the pion mass possible. The determination of the energy levels and the steps leading up to them were performed by the author, the scattering analysis using these energies has courteously been performed by Fernando Romero López using tools already used in Reference [62]. We will introduce this formalism in the following section.

5.1. Formalism

The description of the formalism is mostly based on the work by Blanton, Romero-López, and Sharpe [62]. Equations shown here can be found in that paper.

Three indistinguishable particles can already interact in a non-trivial way without an actual three-particle interaction term. By using the four-vertex interaction $\mathscr{L}_{4\pi}$ term of the effective Lagrangian $\mathscr{L} = \mathscr{L}_{2\pi} + \mathscr{L}_{4\pi} + \mathscr{L}_{6\pi} + \dots$ twice, one can create a three-particle interaction. First particles 1 and 2 interact, then particles 2 and 3 interact, as depicted in Figure 5.1a. As the particles are indistinguishable, all permutations need to be averaged over. This diagram is the leading order (LO) interaction using only $\mathscr{L}_{4\pi}$. Additional such exchanges are possible using the same four-vertex interaction, but do not contribute at LO, which is considered here. The intermediate propagator can go on-shell, creating a divergence. The four-vertex interaction has been studied already and is assumed to be known in the context of three particle interaction.

The simultaneous three-particle interaction comes from the $\mathscr{L}_{6\pi}$ term in the effective Lagrangian and is shown in Figure 5.1b. There we have only a single contact interaction term and no intermediate propagators. The left diagram is subtracted from the whole three-particle scattering amplitude $\mathscr{M}_3$ to make it divergence free, yielding $\mathscr{M}_{3,\mathrm{df}}$. From this an UV-regulated scattering quantity is derived, $\mathscr{K}_{3,\mathrm{df}}$. These are the same at leading order. The higher orders are much harder to derive,



(a) From $\mathscr{L}_{4\pi}$

(b) From $\mathscr{L}_{6\pi}$

Figure 5.1.: Contributions to the three-pion scattering amplitude $\mathscr{M}_3$ at LO. Figure adapted from Blanton, Romero-López, and Sharpe [62].

certain integral equations need to be solved. A discussion of the higher orders can be found in Reference [62, Supplement S2]. It is estimated that the deviations between $\mathscr{M}_{3,\mathrm{df}}$ and $\mathscr{K}_{3,\mathrm{df}}$ could already be 50 % at $M_\pi = 200\,\mathrm{MeV}$, making it hard to connect our lattice results to physical scattering processes.

The three-particle quantization condition can be cast in a form that is similar to the two-particle quantization condition, Equation (4.1). It is given as

$$\det\left(F_3(E,\boldsymbol{P},L)^{-1} + \mathscr{K}_{\mathrm{df},3}(E^*)\right) = 0\,,$$

where F_3 is a function that depends on the two-particle scattering amplitude and geometric functions. As the two pion system naturally is part of the three pion analysis, this function can be assumed as known at this point (details in supplement of Reference [62]). The function $\mathscr{K}_{\mathrm{df},3}$ is an intermediate quantity that depends on the observed finite-volume three-particle energies $E^* = \sqrt{s}$. For such a fit one needs a parameterization. In this work an expansion to linear order in the scattering momentum $\Delta = (s - 9M_\pi^2)/(9M_\pi^2)$ is used:

$$\mathscr{K}_{\mathrm{df},3}(\Delta) = \mathscr{K}_{\mathrm{df},3}^{\mathrm{iso}}(\Delta) = \mathscr{K}_{\mathrm{df},3}^{\mathrm{iso},0} + \mathscr{K}_{\mathrm{df},3}^{\mathrm{iso},1}\,\Delta\,.$$

The linear order does not contain angular degrees of freedom, therefore it is isotropic.

The isotropic approximation to $\mathscr{K}_{\mathrm{df},3}$ can only couple to little groups that are highly symmetric. Therefore for $\boldsymbol{P}^2 = 0$ only the A_1^- irrep and for $\boldsymbol{P}^2 \in \{1,2,3,4\}$ the A_2 irrep exhibit a true three-particle interaction. The other irreps show an energy shift stemming solely from the two-particle interaction. [62, Supplement p. 3]

Using this parametrization of $\mathscr{K}_{\mathrm{df},3}$, together with information extracted from two-pion scattering allows to fit this model to the obtained energy levels and

Ensemble	$1/B_0$	B_1	$M_\pi^2\mathscr{K}_{\mathrm{df},3}^{\mathrm{iso},0}$	$M_\pi^2 K_{\mathrm{df},3}^{\mathrm{iso},1}$	χ^2/dof
cA2.09.48	−0.0481(86)	−1.3(1.1)	0(800)	−200(500)	19.06/(19 − 4)
cA2.30.48	−0.171(22)	−2.0(5)	7500(5600)	−13000(5000)	36.30/(33 − 4)
cA2.60.32	−0.2061(49)	−1.9(2)	4500(1500)	−6200(1800)	58.89/(43 − 4)

Table 5.1.: Results for the combined fit to two- and three-pion scattering with the Adler-2 model in the two-pion sector, using $z^2 = M_\pi^2$ fixed. Table taken from Reference [40].

determine the two parameters $\mathscr{K}_{\mathrm{df},3}^{\mathrm{iso},0}$ and $\mathscr{K}_{\mathrm{df},3}^{\mathrm{iso},1}$. Alternatively one can perform a *combined fit* to the two-pion and three-pion sectors to obtain interaction strengths for both channels in a consistent way.

The expectation from LO ChPT with only S-wave interaction in the two-particle system for the three-particle scattering amplitude is

$$
\begin{aligned}
M_\pi^2\mathscr{K}_{\mathrm{df},3} &= \frac{M_\pi^4}{F_\pi^4}(18+27\Delta) + \mathrm{O}(M_\pi^2/F_\pi^2) \\
&= (16\pi M_\pi a_0)^2(18+27\Delta) + \mathrm{O}(M_\pi^2/F_\pi^2)\,.
\end{aligned}
$$

5.2. Results

The fit of the models of two-pion and three-pion interactions are performed simultaneously. Even if the two-pion interaction were fitted first, the resulting fit parameters would have to enter the three-pion interaction fit in the form of priors to account for the statistical uncertainty. As they become fit parameters this way, one can use the two-pion energy levels directly in a global fit instead of using the intermediate prior construction which only captures the first two modes of the distribution.

The combined fit uses only the A_1 energy levels in the two pion sector as only s-wave interactions are taken into account there. The three pion interaction is modeled with the linear expression, in total we have four fit parameters. In Table 5.1 we summarize the results from the combined fits to each of the three ensembles.

The constant term of the three pion interaction, $\mathscr{K}_{\mathrm{df},3}^{\mathrm{iso},0}$, is positive. This means that the interaction is attractive at threshold ($\Delta = 0$). The results for all three

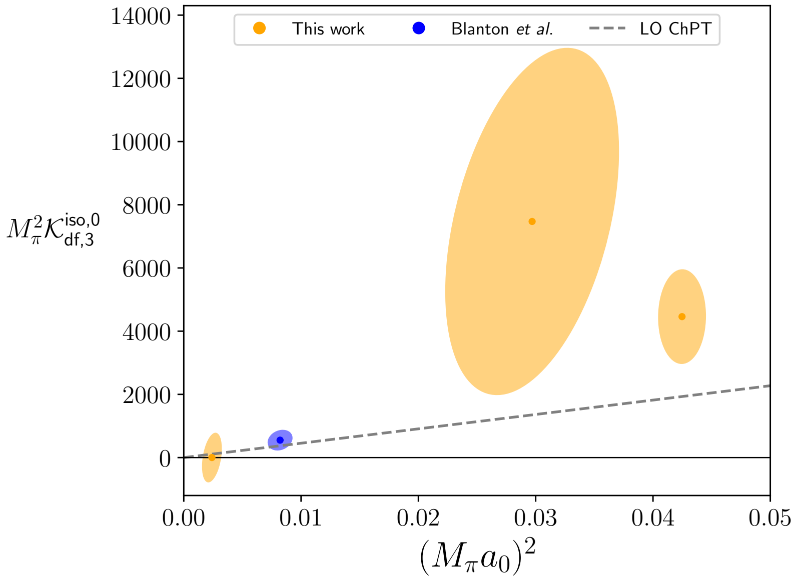


Figure 5.2.: Constant term of $\mathcal{K}_{\text{df},3}$ as a function of the two-pion s-wave scattering length. Comparison with Reference [62]. Figure taken from Reference [40].

ensembles, together with the previous result [62], is shown in Figure 5.2. Our results suggest a larger slope than LO ChPT, which would also be compatible with the previous result. Likely this means that NLO effects are not negligible and the next order of the prediction should be derived at some point.

The linear term, $\mathcal{K}_{\text{df},3}^{\text{iso},1}$ comes out negative for all our ensembles, as well as in the previous work. As one can see in Figure 5.3, this is in stark contrast to the positive values and positive slope that LO ChPT predicts. Here again NLO effects can be very significant. Another source of systematic deviation can arise from the mapping between $\mathcal{M}_3$ and $\mathcal{K}_{\text{df},3}$, where also only the leading order was considered. The potential deviation is estimated at up to 50 % at $M_\pi = 200\,\text{MeV}$ and only increasing from there. For a full treatment of this part one needs to solve the integral equation governing the relation, which was beyond the scope of our work in Reference [40].

Table 5.2 reproduces the results already shown in Table 4.1 together with the combined fit results of Table 5.1. One can see that the results agree very well within their uncertainties. Additionally one can see that the uncertainties have

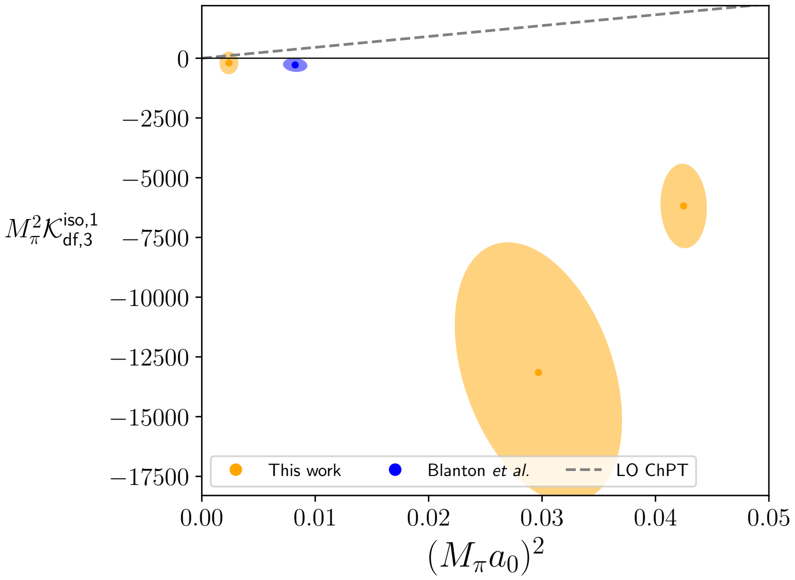


Figure 5.3.: Linear term of $\mathscr{K}_{\mathrm{df},3}$ as a function of the two-pion s-wave scattering length. Comparison with Reference [62]. Figure taken from Reference [40].

	$1/B_0$		B_1	
Ensemble	2π	Combined	2π	Combined
cA2.09.48	−0.0477(90)	−0.0481(86)	−1.4(1.2)	−1.3(1.1)
cA2.30.48	−0.160(23)	−0.171(22)	−1.8(6)	−2.0(5)
cA2.60.32	−0.2090(54)	−0.2061(49)	−2.3(3)	−1.9(2)

Table 5.2.: Results for s-wave fits using the Adler-2 model in the fit to the 2π data alone (left columns) or using the combined fit (right columns). One can see excellent agreement.

been reduced in the case of the combined fit, the three pion interaction is able to further constrain the two pion interaction. We therefore quote $a_0 M_\pi = 0.0481(86)$ as our two-pion scattering length at the physical point. This is compatible with previous results [3, Table 23], but does not constrain them any further.

6. Software development

Deriving equations and conjuring up a data analysis strategy is only part of the story. The other is to actually implement them such that the results are correct, the procedures can be extended and that it runs with sufficient performance to get the work done with the allocated computing time. Scientific codes are often prototypes that pioneer the application of a new method. The method itself will be extended in unforeseen directions and the code needs to adapt. In this chapter we will introduce some key features that made this work possible.

6.1. Parameter value framework

Most people write their data analysis code in a dynamic language like Python or R. These offer easy manipulation of numeric data and creation of visualizations. Usually work groups build up a repository of common functionality, for instance the "hadron" package [80] for R. There is functionality for IO, error estimation, fitting, plotting. Using the provided functions allows to write an analysis of simple pion correlators with just around ten lines of code. A more complex analysis like the one performed for the ρ resonance [30] required a lot more steps, but in principle it just uses all these functions.

Doing variations of steps emerged as a recurring requirement for the analysis. For instance we would use bootstrap as our resampling method and wanted to also see how the jackknife would perform. The removal of thermal states with the weight-shift-reweight method introduces a lot of noise, therefore it is desired to also try it without. Choosing the fit range for correlators is a notoriously ambiguous problem, one needs to try different ranges and see how the fit behaves. The whole analysis had to be carried out for our 15 ensembles, but we started the extraction of the phase shift when the first ensemble was available. Different definitions of the effective mass can yield different insights into the plateau region and remaining unwanted states. There are many more variation points and more of them emerged as the analysis was developed.

Usually one particular set of choices is made and implemented. Senior colleagues will look at the results and ask to have some other choices tried out. The first approach usually is to just change it in the code, re-run and produce another plot. In the best case the other variant will be discarded as useless and the code is changed back. However, there usually will be follow up questions, like to have a single plot with both variants. We now have the problem to have both variants available in the code at the same time. We could let the program run through with the resampling method “bootstrap” and serialize the results into `X_bootstrap.Rdata`, change the program to use jackknife and store the results in `X_jackknife.Rdata`. Then we write a script that loads both data sets and creates the plot. But next the advisor wants to see the energy levels with bootstrap and jackknife. So this ad-hoc changing and re-running of the code does not scale well.

The next approach is to have variables like `X_bootstrap` and `X_jackknife` in the code and write the analysis like that. Everywhere will be variables with the two suffixes, and a lot of code has to be duplicated. Perhaps autocorrelation has not been taken into account and one wants to try different block sizes. This now has various values and one will end up with variables like `X_bootstrap_block1`, `X_bootstrap_block2`, with all the duplication in the parts of the code. Adding another block length would require the introduction of more variables and adding this variant everywhere else. Given the vast number of variation points, the combinatorics quickly spiral out of control.

6.1.1. Embracing variation points

The proposed solution is to embrace these variation points and make them a first class element in the analysis. A fundamental problem with the above attempts is that the there are values within the names of (programming language) variables. One should rather only have them stored *inside* of variables. Unfortunately, programmers gravitate towards deeply nested data structures and arrays of high dimension. All these variants could form a hypercube, the axes describing the variants. Unfortunately most functions do not easily operate on this. Also not all cross combinations are desired, therefore this hypercube will develop gaps. The next approach might be a deeply nested hierarchical structure, but that is only worse.

The concept of “tidy data” [81] solves both issues as the data frames will always be just two dimensional tables. The cost is that some values are replicated, but the additional memory and compute load are clearly worth the predictable structure.

Adding more decision points are just more columns, but the dimensionality of the data structure does not change at all.

The combinatorics of different decision points shall be taken care of by the framework. We want to be able to introduce arbitrary many variants at any point in the analysis. For instance we want to have the following variants for the resampling methods:

`boot_method`
bootstrap
jackknife

Adding a block length is very easy, we just add another column to the this parameter table and obtain the following:

`boot_method`	`boot_l`
bootstrap	1
bootstrap	2
jackknife	1

The long data format allows us to just list the combinations that we want. We do not have to do all, there are no gaps in a hypercube, no need for a nested structure.

We want to allow every part of the code to be done in arbitrary many variations. This unfortunately means that we have to apply our operation to every set of parameters. With the long format it means that we just have to apply the function to each row of this data frame. The presence of another parameter column will not affect the function in any way, it will be transparent to new parameters that have been introduced later.

Each of these variants (rows in the parameter data frame) will have an arbitrary number of variables attached. While operating on a given parameter set the function will have the corresponding values available. After resampling the variable will be the resampled correlation function `corr_boot`. As there are three different parameter sets, the *value* will be a list of three elements, each holding the corresponding `corr`. In the R implementation that is a numbered list where each element is a named list.[1]

In a different part of the code we would have the effective mass. We want to have both the “acosh” and the “log” type to compare them, so we set up the parameter data frame as follows:

`effmass_type`
acosh
log

We want to compute the effective mass for our resampled correlation functions, but these come attached with this data frame of resampling parameters. We have another data frame of parameters giving us the type of effective mass. After doing the combinatorics we will have more parameters:

`boot_method`	`boot_l`	`effmass_type`
bootstrap	1	acosh
bootstrap	1	log
bootstrap	2	acosh
bootstrap	2	log
jackknife	1	acosh
jackknife	1	log

If a parameter occurs in both parameter data frames that are to be merged, an inner product of the values is performed. This could occur if we would want a certain type of effective mass only with a certain resampling scheme.

In the concrete implementation the computation of the effective mass could look like this:

```
f <- function (param, value) {
  effmass <- hadron::effmass.cf(
    value$corr_boot,
    type = param$effmass_type)
```

[1] In Python one can use a `list` containing `dict` elements, in C++ one would use a `vector` each containing a `map`.

```
  list(effmass = effmass)
}

effmass <- pv_call(f, corr_boot, effmass_type)
```

The function `f` accepts one row of the parameter data frame (which is then just a named list) and the corresponding value (also a named list). The return value is a named list containing arbitrary many variables, in this case just the effective mass. Inside the function we use functionality from the "hadron" package to do the actual effective mass computation. The resampled correlation function `corr_boot` is taken from the `value` object. The type of the effective mass is a parameter and therefore comes from the `param` variable.

The function `pv_call` will take one or more of these *parameter-value-objects* and perform the necessary combinatorics. Then the function `f` is applied to each row of the parameters and corresponding values. As the rows of the parameter data frame are independent variants, they are parallel and can be run concurrently. The R implementation does this via multi-processing automatically. The number of variants in a typical analysis easily exceed the number of CPU cores on any machine, therefore effectively utilizing the available resources.

After the above code has been run, the variable `effmass` will be a parameter-value-object containing the parameters `boot_method`, `boot_l` and `effmass_type`, as well as the value `effmass` (same name, different programming language entity). All further steps in the analysis using the effective mass will automatically contain these parameters. In case more parameters are added prior to this code, they will transparently flow through the function `f`. In the above code the parameters `boot_method` and `boot_l` are just ignored.

Should a particular parameter combination be unsuitable, the function `f` could just return `NA` which is a special value in R denoting *not available*. These parameter sets will just be dropped from the result.

Currently only an R implementation of this framework [82] exists, for the lack of a batter name it is called "paramvalf". It consists only of around 350 lines of R code because it draws from the "dplyr" package [83]. If a similar library (like Pandas for Python) is available, porting the framework should be straightforward.

6.1.2. Summaries

This functionality already cures the proliferation of named variables in the program, also it makes extensions easy. There still is a missing piece for the creation of summaries and summary plots. The various values are still independent. As an example we want to plot the effective mass and then visualize the resampling methods and the effective mass definitions. For this we create *summary*, which is a data frame summarizing everything that we want.

```
f <- function (param, value) {
  summary <- data.frame(
    time = 0:(length(value$effmass$t0) - 1),
    effmass_val = value$effmass$t0,
    effmass_err = value$effmass$se)
  list(summary = summary)
}

effmass_summary <- pv_call(f, effmass)
```

By using the specially named return value `summary`, we instruct `pv_call` to flatten the resulting data frame per parameter into one large data frame. The end result will resemble this form:

boot_method	boot_l	effmass_type	time	effmass_val	effmass_err
bootstrap	1	acosh	0	0.9668706	0.01306962
bootstrap	1	acosh	1	1.0456415	0.01215015
bootstrap	1	acosh	2	0.9945144	0.01121410
...	...	...	...	...	...
bootstrap	1	log	0	0.9565278	0.01368002
...	...	...	...	...	...

This format lends itself to visualization with the “ggplot2” package [84] where we could just do the following:

```
ggplot(effmass_summary,
       aes(x = time,
           y = effmass_val,
           color = boot_l,
           shape = effmass_type)) +
  geom_point() +
  geom_errorbar(aes(ymin = effmass_val - effmass_boot,
                    ymax = effmass_val + effmass_val)) +
  facet_grid(~ boot_method)
```

The output is a plot with facets for the different resampling methods, the effective mass each shown with error bars. Different point shapes show the various effective mass types and a color gradient shows the resampling blocking length. In case that this visualization proves ineffective, one can easily choose different visual elements for the variables. Adding another variable like the ensemble, removal of thermal states or different particle channels is straightforward and can also be resolved in this plot. Alternatively one can use the "group by" functionality to create separate plots for these new parameters.

6.1.3. Organizing the code

For instance the analysis for Reference [30] has over 16000 lines in total. In order to retain an overview one needs structure. R provides a serialization format (*RData*) which can store and load every variable, even across different versions and architectures. We have therefore divided the analysis into small chunks that load data, process it and then save it again. The natural interface between these chunks is the saved data. The "paramvalf" package contains the utility functions `pv_load` and `pv_save` which automatically choose a file name based on the variable name. This is orthogonal to the variation points and effectively provides a workflow management for the analysis.

Working on a single chunk is now easy because the number of moving parts is low. The correct execution order for the chunks now is a non-trivial problem, though. With the load and save instructions all the chunks naturally define dependencies onto each other. We just need to run the chunks that *save* a particular variable before we can run one that will *load* it. A Python script provided with this package will parse the R code and generate a directed acyclic graph (DAG) from this. The graph is visualized with GraphViz [85] and also emitted as a Makefile. Figure 6.1 shows a partial data flow diagram for Reference [30]. Calling `make` via a wrapper script will then execute all the chunks of R code such that everything is consistently updated.

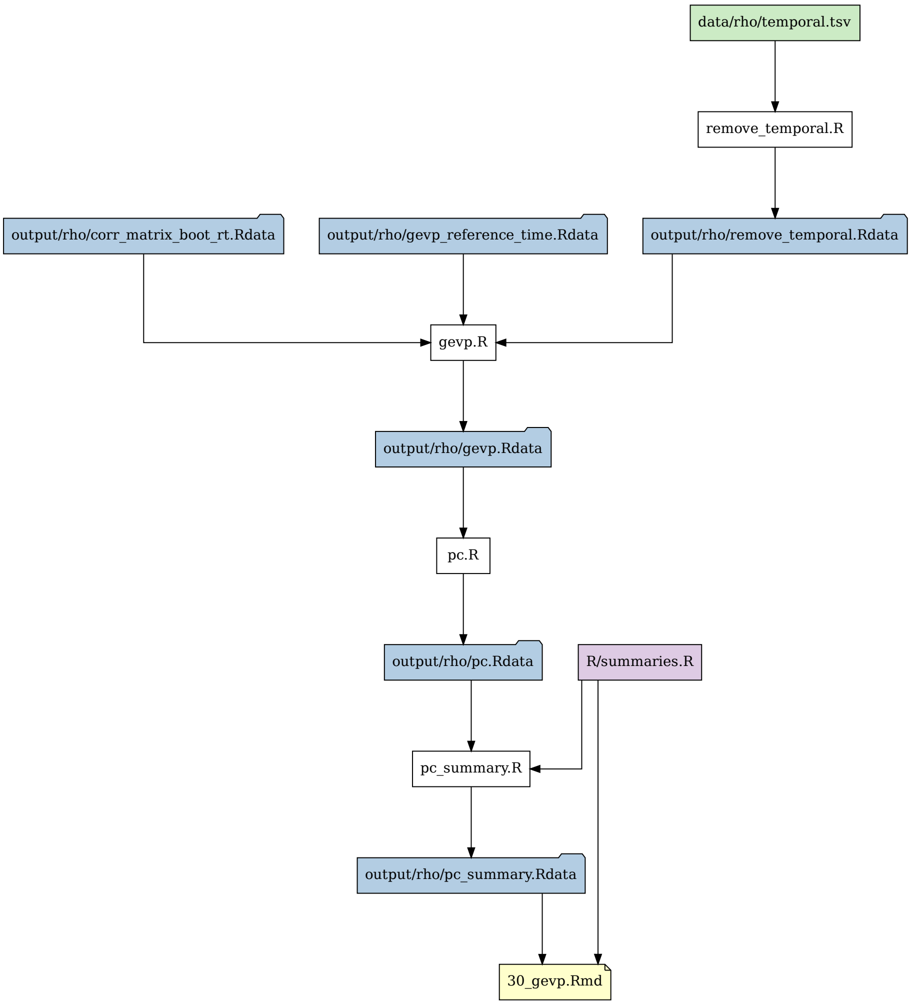

data/rho/temporal.tsv
remove_temporal.R
output/rho/corr_matrix_boot_rt.Rdata
output/rho/gevp_reference_time.Rdata
output/rho/remove_temporal.Rdata
gevp.R
output/rho/gevp.Rdata
pc.R
output/rho/pc.Rdata
R/summaries.R
pc_summary.R
output/rho/pc_summary.Rdata
30_gevp.Rmd

Figure 6.1.: Partial data flow graph from Reference [30]. White boxes are R code chunks, green boxes mark text files with parameters. Blue boxes denote serialized data and purple boxes are function files that contain utility functions within the analysis. Yellow chunks are R Markdown documents that generate a report.

This part is independent of the actual parameter-value constructs. In principle one could split this into two different libraries. The memory usage scales linearly with the number of variants that are performed, therefore there have been plans to automatically serialize data as it is computed. So far this has not been implemented satisfactorily, but it could couple both parts tighter at some point in the future.

6.1.4. Converting parameters to values and back

The values from different parameter sets are processed always separate from each other. This is fine only in most of the cases. We routinely run an analysis on ensembles with different lattice extents L/a, different pion masses aM_π and different lattice spacings a. Several analysis steps (resampling, GEVP, fitting) are done per ensemble. Extrapolations however require that we take several ensembles and regard them as a whole group. One concrete use case is analysis of finite size effects. We want to make groups that have the same lattice spacing and pion mass, but different lattice extents. In SQL terminology we want to *group by* lattice spacing and pion mass. Actually we also want to group by everything except the lattice extent. In the `pv_call` function there is a parameter called `convert` which takes the names of the parameters that shall not be grouped and rather *converted*.

As an concrete example one can use the effective mass data from before. There we had the parameters `boot_method`, `boot_l` and `effmass_type`. If one were to compute whether the means and errors are significally different for a specific time slice (Z-test), one would need to group by `boot_l` and `effmass_type`, but would want to let go of `boot_method` as an external parameter. The following table shows the four groups that are generated.

`boot_method`	`boot_l`	`effmass_type`
bootstrap	1	acosh
jackknife	1	acosh
bootstrap	1	log
jackknife	1	log
bootstrap	2	acosh
bootstrap	2	log

Some of these groups do not contain both resampling methods. Therefore we could not do a comparison with them and have to return `NA` instead. The meaning of `param` and `value` that are passed to the inner function will change. `param` will instead be a data frame with multiple rows (two or one in this example) containing the columns `boot_l` and `effmass_type`. The `value` will contain the fields `boot_method`, a list of strings; and `effmass`, a list with the `effmass` values corresponding to the rows in the `param` data frame. The function needs to then pick out the two effective mass objects and perform the comparion. The result will have the parameters `boot_l` and `effmass_type`, and it will only have two rows as we only keep the first two groups in this example.

The opposite direction is also possible. When solving a GEVP, we have one $n\ times n$ correlator matrix going in per parameter combination. The output however has a *principal correlator id*, which we ideally expose as a new parameter. For this the inner function needs to return a list with just one entry named `paramval`. It must be a parameter-value-object containing just the new `correlator_id` parameter. The `pv_call` function will recognize that and resolve the nested data structure into a large parameter-value-object with many more rows and the additional parameter.

6.1.5. Shortcomings

Although this framework has served very well in constructing the ρ and 3π analyses, there are a few shortcomings which currently lack a proper solution.

The framework neatly decouples all the various parameters from each other, making it very easy to add more parameters and also more steps in the analysis graph. Adding more data is very easy as well, one just makes the parameter tables longer. Dependencies will get automatically resolved and in the end everything is updated.

Memory usage Having the values for all parameter combinations in a single object can lead to very large memory requirements. Storing the effective masses and their bootstrap samples for over ten ensembles, tens of energy levels per ensemble, multiple fit methods and two resampling methods can require around 10 GB of memory. Usually multiple such objects (say effective masses, fits and principal correlators) need to be kept in memory. For the analyses in this work the memory requirements are such that they cannot run on a regular workstation any more. Due to the computing complexity it is advisable to use a larger server anyway, but the memory requirements force one to do so.

One solution attempt has been implemented but did not catch on: Instead of having the *values* for all parameter combinations in memory, they can be serialized into individual files and loaded as needed. This way one would only need as much memory as there are processes working on the data. Memory scaling would be constant in the number of parameter combinations and not linear. The caveat is that there is a lot of disk IO with small files. There must be a threshold, otherwise the overheads would become crippling for small objects.

The largest issue is ensuring integrity of the files. The `paramval` object will only contain references to other files in the `value` section. These could be stored in files with random names, but then it would be hard to determine which files have expired. The disk usage would grown with each execution of the analysis, which is not acceptable. Instead it was tried to create a directory with numbered files. When the object is created again, the directory is pruned first and then each value is serialized again into a numbered files. This way old files are automatically deleted and the directories are organized and human serviceable.

But how does one know the name of the variable that the result will be stored in? In normal code, it looks like this:

```r
result <- pv_call(func, data)
pv_save('project', result)
```

The function `pv_call` cannot know that the result will be stored in a variable called `result`. In order to circumvent this, the function signature was changed to `pv_call(result, func, data)` and used R's non-standard evaluation [86] to create a variable called `result` in the calling scope. This way of writing is just a bit inconvenient but still possible.

Experiments with the described changes have been performed and were underwhelming. R is a managed language and uses a garbage collector. Even calling `gc()` in the code will not necessarily free up memory. Also loading data in the `pv_call` function had to be done carefully because the data would otherwise accumulate during the `pv_call` function call and use as much memory as it did before. Actually it would use more because of R's *pass by promise* automatic read-only referencing which helps to lessen the memory footprint. Multiprocessing also increases the memory usage.

In the end it was decided that all the additional trouble was not worth it. With the changes the analysis could in principle run on a laptop, but there it would take very long due to the low clock speed and core count. As the cluster frontend has enough memory anyway, these changes have been reverted and large analyses just require memory in the ballpark of 50 GB.

Dependency granularity Another problem is that the framework tracks dependencies only on a file level, which means per PV object. Say we just have a single fit range for a particular energy level. Then we would expect that only that fit would be performed. Unfortunately the granularity of change tracking is on a file level and the text file with the fit ranges was changed, so *all* fits will be re-done, although the results will be exactly the same as before.

One could just take some hash digest of all the parameters and data that go into a result and store that with the results. If the input files have changed, it could be checked whether the hashes have not changed and the previous output would be used, otherwise it would be computed anew. It is rather problematic to get the dependencies right in all cases, for instance there could potentially be inconsistent results that are computed using different versions of installed libraries. This is not tracked in the present state either, but at least all the data in one PV object will be consistent.

This is harder than it might appear at first. If not all dependencies are tracked correctly, results would be inconsistent, users lose trust and just re-run everything. So far attempts to tackle this problem have been held back because the fear of inconsistent states has been larger than the need for more efficiency.

Debugging New abstractions require new tools for debugging. The additional abstraction here is over variants. When any of the variants fail, the whole function call of `pv_call` fails. One can set a global boolean variable `debug_mode` which will force all `pv_call` invocations to use serial execution using `lapply` instead of parallel execution via `pbmclapply` [87, 88]. This removes the multiprocessing problem from the debugging but leaves the debugging inside the `lapply`. One needs to run until the variant with the error comes up and then debug it. Just calling `browser()` inside the closure will not be of much help as succeeding variants need to be skipped.

The best option currently is to set the following:

```
debug_mode <- TRUE
options(error = browser)
```

All `pv_call` will then use serial execution with the main process. Errors (as triggered via `stop()`) then trigger the debugger and one can investigate the faulty variant. Investigating the `param` row in the closure parameter gives clues to which variant has failed.

It would be desirable to get to errors more quickly and also be able to debug parallel execution of variants. So far there are no fruitful plans to achieve this goal.

6.2. Projection code

One part in the toolchain is a projection code which implements Equation (2.2). For Reference [30] we have used the *sLapH Projection* code [89]. It takes numeric correlation functions and projects them into the lattice irreps that couple to $I = 1$ and $j = 1$. Unfortunately it is a rather monolithic code that had analytic simplifications done before the implementation. Everything was tailor made to the specific channel and therefore extension was deemed unsuitable.

In order to prevent a similar situation in the future, the stated goal of the new code is set to support arbitrary many particle in every partial wave and isospin. Although only pions were going to be used at first, extension points for non-scalar particles have been left in the code. All analytic simplifications that arise from a specific isospin and spin channel shall only be done within the code. The need of analytic algebra led to usage of the *Wolfram Language* [90], the language in the *Mathematica* computer algebra system. Unfortunately the free software alternatives were not deemed sufficiently capable. Additionally the *Quark Contraction Tool* [91] was already available for the Wolfram Language, alleviating the need to program Wick contractions.

The new projection code [36] is built in a modular fashion such that the various steps are somewhat independent of each other. The repository contains many dozen pages of documentation explaining the design decisions, implementation details and also some usage instructions.

6.3. Contraction code

For the three pion contractions the existing *sLapH contractions* code [92] had been extended to support more than two particles. This alone would not have sufficed as the number of momentum combinations with three particles at source and another three at the sink is significantly larger than with two particles. Without significant changes the code would have taken way too much computer time and memory and could not even fit onto compute nodes with 96 GB as they are deployed on JUWELS [39].

In a round of refactoring a caching structure has been introduced. This allowed the reuse of elements that have been already computed while still being able to control the size by clearing the cache. Quantities that have been computed eagerly were then computed on demand and cached. The disadvantage has been large memory usage, which did not pose a problem for two particles at the time for Reference [30]. Each thread on the CPU had it own cache, letting it work completely independent from the other ones. The many source-sink time slice combinations were just distributed to the threads.

For three particles another round of refactoring was needed, partially undoing changes from the preceding round. So far only elements of the form $\mathrm{Tr}(QQ)$ had been cached, which makes computation of `C4cD` and `C4cV` diagrams very cheap. The diagrams consisting of four quark lines, `C4cB` and `C4cC` however did not profit from this as they are of the form $\mathrm{Tr}(QQQQ)$. There have been some tricks that cache the intermediate QQ objects to essentially bring it down to the computation of all QQ objects and then assembly of the trace as $\mathrm{Tr}((QQ)(QQ))$ from cached objects. The diagrams for three particles on either side will have $\mathrm{Tr}(QQQQQQ)$ and therefore the intermediate QQ objects needed to be cached to facilitate reuse across all diagrams that are computed.

The sheer number of momentum combinations however made it feasible to just hold all the intermediate objects in memory for a single thread. Usage of more threads meant a multiple of the memory. At the time the code could only run on machines which had around 60 GB of memory per thread, which is completely unobtainable at the moment. The target machine, JUWELS, has 48 CPU cores and 96 GB of memory per node, giving each thread just 2 GB. Using only few threads would have been a giant waste of resources as they are billed by the full node. We have decided to not parallelize over the time slice combinations but rather over the momentum combinations. This way the memory usage becomes virtually independent of the number of threads used, making it a much better fit for existing supercomputer system.

Momentum combinations that get computed were generated by the code during startup on each configuration. For at most two particles at either source or sink the number of combinations is limited and the existing brute force algorithm finished in a few minutes. For a two-to-two particle diagram it would take all possible single particle momenta up to $\boldsymbol{p}_i^2 \leq 4$, perform all combinatoric combinations of two of particles and only then check whether total momentum was as desired. It would have been much simpler to just subtract the momenta from the total momenta to obtain the momentum of the last particle. This algorithm did not scale for three particles at source or sink, which is not surprising in hindsight.

As the code was rather convoluted and the algorithm inefficient, it was decided to just use the list of correlators demanded from the projection code as input. The user will have to create the projection prescriptions before being able to run contractions. This way only correlators are generated which will be used in the projection later on. In the old state certain correlators were computed but never used, another waste of resources.

This fundamental change in the usage also allowed us to get rid of the seemingly arbitrary momentum cutoffs that had been in place. The rules for the individual particle momenta have been the following:

$$\boldsymbol{p}_i^2 \leq 4 \,, \quad \sum_i \boldsymbol{p}_i^2 \leq C(\boldsymbol{P}^2) \,,$$

with the cutoff function given as

$$C(0) = 4 \,, \quad C(1) = 5 \,, \quad C(2) = 6 \,, \quad C(3) = 7 \,, \quad C(4) = 4 \,.$$

These arbitrary seeming cutoffs serve the purpose of avoiding states which have their non-interacting energies way above the $4M_\pi$ threshold. This heuristic is no longer needed with the new projection code as its modular fashion also allows to export a listing with all the individual particle momenta coupling to a given irrep. Figure 3.1a shows this for a typical case. Then one can insert the lattice extent L/a and the pion mass aM_π to decide which states lie above the relevant thresholds. Usually one operator above the threshold is kept to stabilize the GEVP, all higher operators are just cut in order to save computing resources. The generated list of actually useful correlation functions is then used by the contraction code to compute them.

All the refactoring could have introduced various errors in the program. In order to prevent this, a full end-to-end integration test has been introduced before touching the code. In this way any deviations in the end result would be spotted quickly. As a contraction on a real lattice would need too much time, a special small 4^4 lattice has been generated for testing. The tests for all diagrams run within a minute on a regular computer. Using the service from Travis CI [93] they are run each time a new commit is uploaded to GitHub. While making changes to the code, the tests were never allowed to fail, guaranteeing that the results still match the initial version of the code. A few normalizations have been corrected carefully, this lead to a controlled update of the reference data. Diagrams which were newly added did not have a reference, there a cross check was done with an independent implementation.

Going forward, the next projects can benefit from an improved contraction code that is ready to be extended with more diagrams stemming from lower-than-maximum isospin channels.

7. Conclusion

In this work we present the first computation of the three-pion contact interaction at physical pion masses. The constant term is in agreement with LO ChPT and previous data, whereas the linear term shows a trend into the wrong direction.

We obtain $a_0 M_\pi = 0.0481(86)$ as our two-pion scattering length at the physical point using the two and three pion spectra in a combined fit. This is compatible with previous results, but does not constrain these any further.

Going forward, one could derive the NLO ChPT prediction for 3pi and properly solve the integral equations connecting the unphysical $\mathscr{K}_{\mathrm{df},3}$ of the quantization condition with the physical $\mathscr{M}_3$ such that a comparision far away from threshold becomes more meaningful.

The statistics on the cA2.30.48 ensemble is rather low. Especially in the three pion scattering one can see that it has the least statistical weight. A doubling of the statistics is possible and could be done in the future to improve significance of the results.

The ETMC has a group of ensembles with $N_\mathrm{f} = 2 + 1 + 1$ non-clover twisted mass fermions, which would offer additional flavors but lacks an ensemble at the physical pion mass. Therefore these ensembles were not used for this work. A new group with $N_\mathrm{f} = 2 + 1 + 1$ clover twisted mass fermions is in the making and could be used to investigate κ-physics in the context of three-pion scattering while still benefiting from the $\mathrm{O}(a^2)$ improvement by the clover term. Additionally this ensemble group will feature multiple lattice spacings, such that the size of lattice artifacts could be quantified.

As the RFT formalism has been extended to lower-than-maximum isospin channels [77], numerical studies could follow with reasonable effort using existing software. The new projection code [36] has been kept general enough that an extension to other isospins is not necessary, but spin would need to be finished such that intermediate ρ resonances could also be probed with operators. The contraction code [92] is also refactored such that adding the required quark-disconnected diagrams is not much more work and fits into existing structures. The energy extraction of the correlators does not depend on the isospin channel and could just be used directly. The changes in the RFT formalism would have to be implemented.

Finally a few details about the software engineering needed to realize the physical results have been given. The parameter-value-framework was introduced, it provides flexible variation points in the analysis as well as a complete workflow management to guarantee a consistent state of the analysis. This framework is now used in multiple other projects. A whole new group theoretical projection code has been created, it can be extended to provide projections for other physical channels and particles with spin. The changes to the contraction code have added features, made it faster and reduced the memory usage. Future extensions for more physical channels will be rather straightforward.

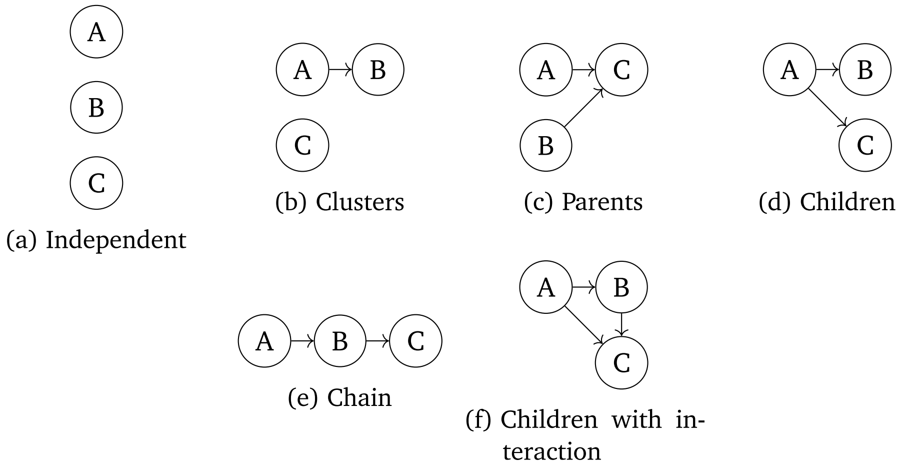
A. Directed acyclic graph Markov chain Monte Carlo

“Correlation does not imply causation” is a common statement to prevent people from trying to read too much into a correlation [94]. Given samples for the random variables A, B and C one can compute the correlation between them and make hand-waving conjectures about their causal relationships. The problem is that with limited data one is prone to fall for spurious correlations of which there are many amusing examples [95].

Taking a Bayesian viewpoint however, one can give probabilities for different causal structures. Figure A.1 shows the possible different ways that the variables could actually interact with each other, assuming that there are no cyclic relationships. These graphs are *directed acyclic graphs* (DAG). In particular Figure A.1d shows how the random variables B and C could have large correlation but no direct causal contact.

The interaction between the graphs needs to be modeled with some function, having some edge parameters. A probability density function (PDF) depending on the values of the random variables and the edge parameters then gives a likelihood for the given values in that particular graph. By sampling this PDF one will obtain samples for the random variables. All possible graphs need to be sampled this way to map out the PDF. Once this has been done one can use the Bayesian approach to invert the relationship and give a likelihood for a particular graph (with certain edge parameters) given samples of the random variables. The hope is to infer the causal structure just from the data, exactly as it is commonly pointed out to be a fallacy to try so.

In this chapter we will work with this approach and attempt to test the approach with increasingly difficult models. There are a lot of unknowns, the model dependence is a key factor. In reality any sort of non-linear functional dependence could arise. Additionally the restriction to acyclic graphs could be too strict to explain certain phenomena. Still it is worthwhile to try this with a toy model.



(a) Independent

(b) Clusters

(c) Parents

(d) Children

(e) Chain

(f) Children with interaction

Figure A.1.: All possible graph topologies for the causal relationship between the three random variables A, B and C.

A.1. Model

The graph has p nodes and q directed edges that connect the nodes. The number of random variables X is also p, each node corresponds to a random variable. We have N_{samples} samples of each random variable X_i as the vector $\boldsymbol{x}_i$. Not all nodes need to be connected, some can be multiply connected. Some of the nodes have no parents, they are called *exogenous*. Nodes with parents are called *endogenous*. Their count is N_{endog}. The node numbers of the parents of node i are denoted as P_i.

The interaction from one variable with the next is modeled with a logistic curve. Each node has two parameters (α and σ), each edge also has a parameter (β). The node parameters are indexed with a node number i, the edge parameters with parent p and target node number i as $\beta_{p \to i}$.

The PDF $f(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\sigma}; \boldsymbol{X})$ is computed from these parameters as follows. We have a simple logistic function,

$$\text{logistic}(z) = \frac{1}{1 + \exp(-z)} .$$

that serves as the source of non-linearity in the model. Various other functions are possible.

The exogenous random variables are taken to be normally distributed. We use the logarithmic form of the normal distribution,

$$\log\operatorname{norm}(x,\mu,\sigma) = -\frac{(x-\mu)^2}{2\sigma^2} - \frac{1}{2}\log(2\pi\sigma^2)\,.$$

The expected mean value for a certain random variable i on a given sample n is computed from the parameter α_i intrinsic to that node together with the edge weights and the parent samples:

$$\mu_{ni} = \operatorname{logistic}(\alpha_i + \sum_{j\in P_i} \beta_{p\to i} x_{np})\,.$$

The logarithm of the PDF is then computed as the sum all the log likelihoods:

$$\log(f) = \sum_{i=1}^{N_\text{endog}} \sum_{n=1}^{N_\text{samples}} \log\operatorname{norm}(X_{ni}, \mu_{nip}, \sigma_i)\,.$$

The model is formulated such that the data is *standardized* to have zero mean and unit standard deviation. This process is usually used for meaningless arbitrary scales like the ones ranging from “very bad” to “very good” on questionnaires. In physical applications there usually is an absolute scale which should not be standardized away. The model would need to be extended with additional parameters for the original mean and standard deviation such that absolute scales could be recovered. During the conducted experiments it was found to be far less stable than the standardized model.

In a realistic application the samples for the random variables would be given and both the graph and the parameter would be inferred. For the development of the method a fixed graph and fixed parameters are used. From these the samples are created by first sampling the exogenous variables from a normal distribution with mean zero and width σ_i. The endogenous variables are sampled from a normal distribution with mean μ_{nip} as computed in the above prescription. The algorithm will then be given the graph must then be able to infer the fixed parameters.

A.2. Implementation

For sampling a PDF with many parameters the lattice practitioner directly thinks of the hybrid Monte Carlo algorithm [9]. The chosen model has continuous parameters and is differentiable, therefore the algorithm is applicable. The molecular dynamics part needs an integrator. At first the simple *leap frog* algorithm has been used, in the end the minimal norm integrator of fourth order [96] has shown much more efficient.

The model function has been implemented in Python using the SciPy [97] matrix and scientific function libraries. The analytic gradient is computed automatically using the *autograd* [98] library. It uses the concept of *automatic differentiation* and introspects the algebraic expressions and uses appropriate differentiation rules. This concept allows to quickly change the model without having to derive and implement the gradient. In machine learning these tools are in widespread use, whereas lattice QCD has only few operators and manual differentiation is still feasible. This convenience has a price in computation cost and will feature in Section A.4.

Input and output has been done exclusively with the YAML serialization language [99]. It is human and machine readable, also libraries for all relevant programming languages exist. Analysis of the data was therefore easily done with either Python or R. For larger numerical data the text format takes up much more storage space, takes much longer to read and write and can suffer from rounding issues if not enough significant digits are used in the output. The Python YAML writer automatically serializes sufficiently many digits and does not have this problem.

Unfortunately the source code of the DAG-MCMC project [100] is still private.

The HMC implementation was tested first by sampling from a normal distribution with logarithmic PDF of just $-x^2$. A second verification was a simple χ^2 fit which can be compared to a bootstrapped fit. Typical quantities to check are the values and standard errors of the various fit parameters, their autocorrelation time and the distribution of $\exp(-\Delta H)$ of the updates. Also the scaling of $|\Delta H|$ with the MD integration step length δt showed the typical quadratic scaling that one would expect for a second order integrator. The computation of the forces has been done with the autograd implementation, just like for the model of interest. After these tests with known reference results have been passed, we had confidence into the HMC implementation.

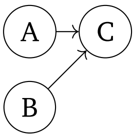


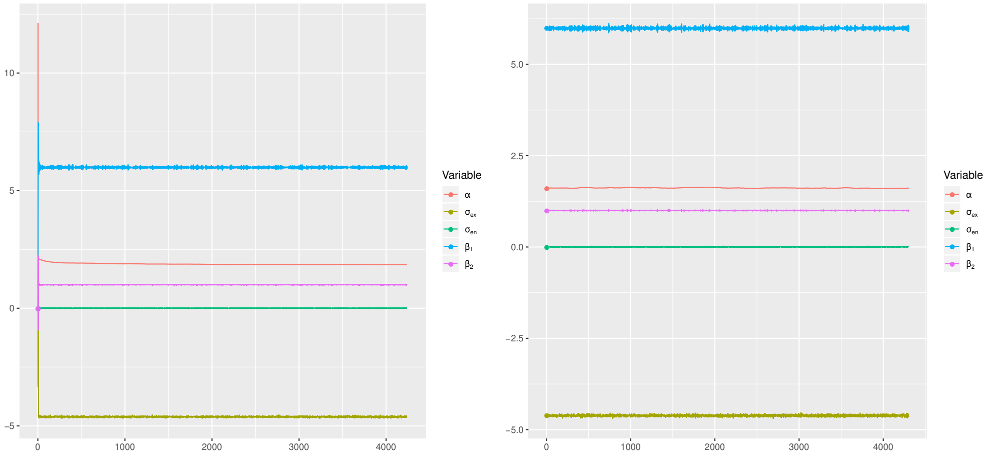
Figure A.2.: Simple collider model

A.3. Results

At the beginning it was unknown which graph complexity would be suitable for the first test runs. Therefore larger and smaller models were tested. The simplest non-trivial model is one with a single endogenous variable with two parents. It will receive input from two external variables and they become mixed. Figure A.3 shows the graph for this model. There are two edges. In total this model has five variables: α, σ_{ex}, σ_{en} and the edge weights β_1 and β_2.

With suitable low noise levels the system can indeed figure out the actual values of the system. These are the values of the five variables that have been used to generated the data samples that the DAG HMC works with. In Figure A.3 there are the values of the five variables plotted against the HMC update number. The HMC should be able to find the global minimum of the log-likelihood function; in real applications this might take a very long time during which the algorithm gets stuck in a local minimum. In order to find out whether the starting conditions have an impact on the resulting equilibrium the simulation has been started with all variables set to zero and also with all variables already at their true value. The latter case is cheating for our purposes as we do not known them for real applications. However both starting values seem to yield the same equilibrium rather quickly.

This simplest model also gave us the opportunity to experiment with the acceptance rate. Depending on the model the target acceptance rate, but in general there is a sweet spot somewhere between 50 % and say 90 %. A too high acceptance rate means that one does not do much progress in each update, leading to high autocorrelation times. The number of accepted measurements might seem high, but the number of uncorrelated measurements is rather low. If the acceptance rate is too low, there will not be any progress and the required number of raw updates is again high. The program monitors acceptance rate during the simulation. When it is too low the number of integration steps will be increased by a constant factor. This lowers δt and therefore also $|\Delta H|$. All updates leading to that change need to



(a) Start from zero

(b) Start from known values

Figure A.3.: Evolution of the variables α, σ_{ex}, σ_{en} and the edge weights β_1 and β_2 with the Markov chain updates. The difference between the two histories are the starting values (denoted with circles at update 0). The left one starts with all variables at zero, whereas the right one starts from the known values. The simulation is run with $\tau_0 = 0.1$ and $N = 50$ steps in the MD part.

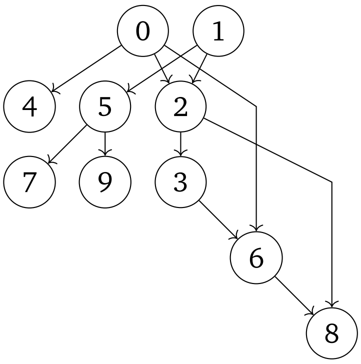


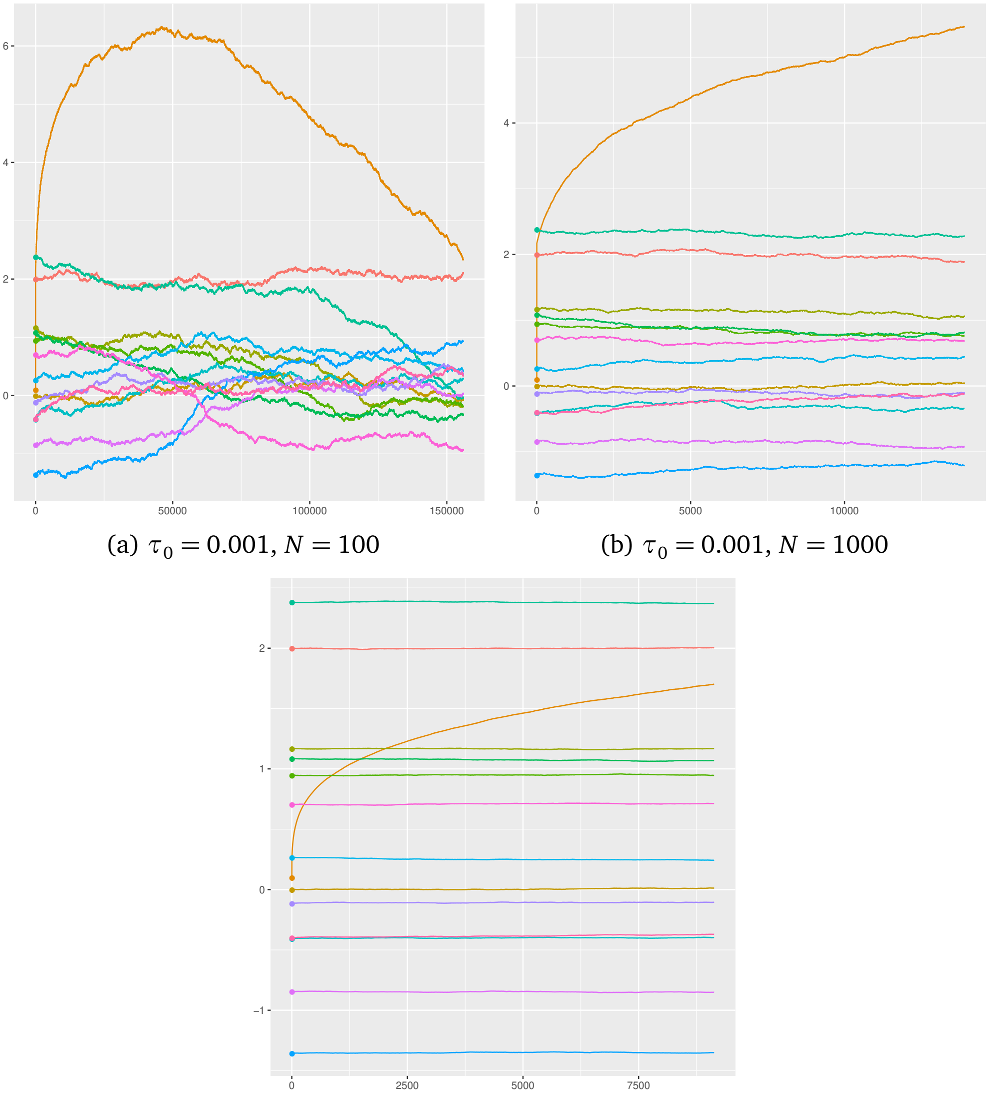
Figure A.4.: Model with 10 variables.

be discarded as the ergodicity is violated with a change in simulation parameters. These automatic changes usually happen early on until a suitable acceptance rate is stabilized.

A more complicated model that was chosen has 10 variables. Two variables are exogenous, all the others are connected such that each endogenous variable has between one and two parents. The generated graph has 11 edges, as shown in Figure A.4. In total the model has $3+11 = 14$ parameters that need to be measured via the HMC. In order to test the stability of this model we have started the runs from the known central values.

The MD integration length τ_0 and the number of integration steps N can be tried to cover various orders of magnitudes. When the integration is performed too coarsely the forces will become very large or even diverging and spoil the simulation. Shortening τ_0 and/or increasing N will make the system more stable in most cases. There has been a very curious case where the σ_{ex} would simply diverge even when started from its known central value, see Figure A.5a. Increasing N by a factor of 10 does not cure this problem, it will just make the simulation progress much slower that not enough statistics could be gathered in reasonable time. See Figure A.5b for the comparison. This initial divergence remains even when one also decreases τ_0 by another factor of 10 as can be seen in Figure A.5b. The simulation just becomes so slow that waiting for it to come back down would not be feasible.

We would have liked to investigate this further by just letting the simulation run longer. The simulation as implemented only runs on a single CPU core and the



(a) $\tau_0 = 0.001$, $N = 100$

(b) $\tau_0 = 0.001$, $N = 1000$

(c) $\tau_0 = 0.0001$, $N = 1000$

Figure A.5.: Simulation of a more complicated model with 10 variables and 11 edges. The diverging parameter is σ_{ex}.

autograd implementation is rather slow. It was deemed out of reach until the performance was improved significantly.

Reparametrization of the σ parameters as $\sigma = \exp(s)$ to force them to become positive did not change the strange behavior that we have seen. Using the fourth order integrator made it better, yet still progress has been too slow to properly analyze the model.

More complex models, with 30 variables, were also tried. There the autocorrelation time for some parameters grew to figures like 500 iterations. Given the overall slow progress in the MD, this meant that the number of usable updates was way too small to properly sample these types of graphs yet.

The extended model that can work with non-standardized data was too unstable to yield sensible results with the given performance of the program. Therefore this branch of investigation has been postponed until after the performance is improved.

A.4. Performance issues

The automatic differentiation is very nice as it lets the user change the model and directly access the needed gradient for the HMC. Unfortunately the concrete *autograd* implementation in Python running on cpython is the dominating bottleneck of the application. Figure A.6 shows a profile of a HMC run. The top layer is the benchmark function `task`, then comes the HMC driver function `do_hmc`. In the third line one can see that virtually all the time is spent in the integrator `omf4` whereas only a sliver is spent in other parts of the code. The integrator again spends most time in autograd functionality (`nary_f`).

There are two steps in the computation of the gradient. First the function is evaluated using special objects that behave like NumPy arrays but built an expression tree with all the operations that are applied to them. This slows down the actual evaluation significantly. The standard rules of differentiation are applied to these expressions. A junction with a product will invoke the *product rule* and a function evaluation the *chain rule*. This way an analytic expression of the gradient it built up. In a second step concrete values are inserted into this gradient expression tree and is evaluated bottom up. Autograd cannot tell if the structure of the gradient will change when it is evaluated with a different function argument and therefore has to re-build it every single time.

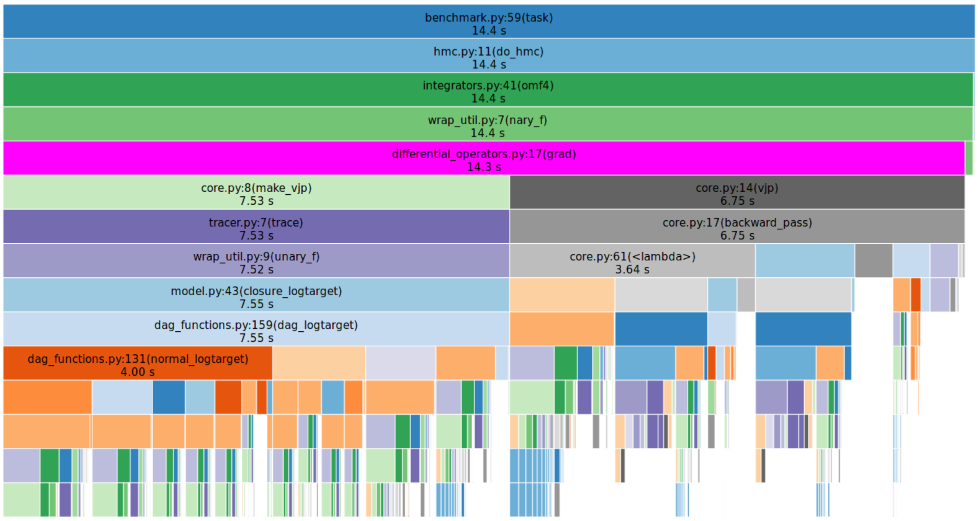


Figure A.6.: Runtime profile of a HMC run showing the dominating cost of the gradient function `grad` (marked in pink). Setup (left side) and evaluation (right side) have roughly the same cost. The profile has been visualized with Snakeviz [101].

Fortunately there is a concept of a *constant graph* where the user can specify knowledge about the structure. With our model we know for sure that the structure will now change, therefore the graph can be generated once and then become frozen for the remainder of the program. The effect of this can be seen in Figure A.7 where the evaluation part has been reduced. Unfortunately this saves less than 50 % of the costs, so the program is not the needed orders of magnitude faster.

One general way to make a Python program perform faster is by using the Numba [102] *just in time* (JIT) compiler. One just lets it turn the Python code into machine code using the LLVM backend. Unfortunately this does not work for every data type and every operation. The default settings just compile what they can. In our case there was no performance benefit. Forcing Numba to compile everything just yielded in errors about unsupported access mechanism to arrays. Therefore we would have to change all the code such that it would support Numba.

The autograd library is not being developed further, activity has gathered around JAX [103], which combines automatic differentiation with JIT. This way the analytic expression for the gradient is computed once via special expression objects in Python. The generated code will then hopefully have native machine speed and

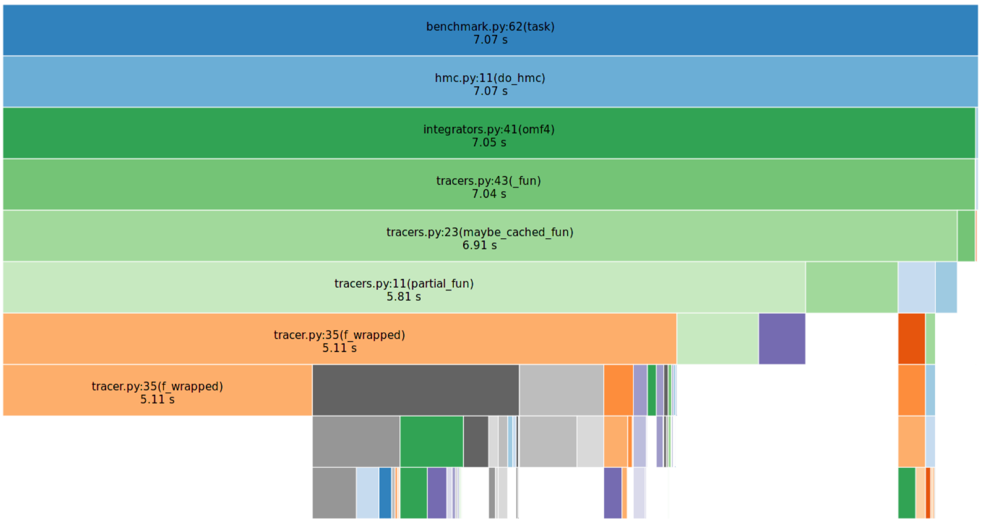


Figure A.7.: A profile similar to the one shown in Figure A.6, just with the constant graph optimization. One can see how the evaluation cost (right side) is much smaller than the setup cost (left side) now.

could in principle even be used for GPU. For the most part the JAX library can be dropped into an existing autograd project. However some NumPy operations, especially slice assignment and cumulative sums, were supported by autograd but not JAX. Therefore the introduction of JAX would have required significant changes to the code. Instead of these changes we decided to rather directly aim for writing the model and simulation in machine code.

A.5. Outlook

The project cannot advance to larger systems or more statistics or non-standardized models without significant performance improvements. Therefore the next step would be to implement the model PDF in a computer algebra system like Mathematica and compute the gradient analytically once and for all. This expression could then be implemented in C++ and become native code.

Additionally we would gain the flexibility to evaluate single components of the gradient independent of each other. The way that autograd lets us access that is only the full gradient. Computing a single component is as expensive as computing the whole gradient. Integrating different components on different time scales therefore does not give any performance advantages in the current state.

B. Long tables

The following table contains the individual momenta that couple to given total momentum $\boldsymbol{P}^2$ and irrep Γ in the two-pion sector.

$\boldsymbol{P}^2$	Irrep	$\boldsymbol{p}_2$	$\boldsymbol{p}_3$
0	A1g	$(0,0,0)$	$(0,0,0)$
0	A1g	$(0,0,-1)$	$(0,0,1)$
0	A1g	$(-1,-1,0)$	$(1,1,0)$
0	Eg	$(0,0,-1)$	$(0,0,1)$
0	Eg	$(-1,-1,0)$	$(1,1,0)$
1	A1	$(0,0,2)$	$(0,0,-1)$
1	A1	$(0,0,1)$	$(0,0,0)$
1	A1	$(-1,0,1)$	$(1,0,0)$
1	A1	$(-1,-1,1)$	$(1,1,0)$
1	B1	$(-1,0,1)$	$(1,0,0)$
1	B2	$(-1,-1,1)$	$(1,1,0)$
1	E	$(-1,0,1)$	$(1,0,0)$
1	E	$(-1,-1,1)$	$(1,1,0)$
2	A1	$(1,1,0)$	$(0,0,0)$
2	A1	$(1,1,-1)$	$(0,0,1)$
2	A1	$(1,0,-1)$	$(0,1,1)$
2	A1	$(0,2,0)$	$(1,-1,0)$
2	A1	$(0,1,0)$	$(1,0,0)$
2	A2	$(1,0,-1)$	$(0,1,1)$
2	B1	$(1,1,-1)$	$(0,0,1)$
2	B2	$(0,2,0)$	$(1,-1,0)$
3	A1	$(1,1,1)$	$(0,0,0)$
3	A1	$(1,1,0)$	$(0,0,1)$
3	A1	$(0,2,0)$	$(1,-1,1)$
3	E	$(1,1,0)$	$(0,0,1)$
3	E	$(0,2,0)$	$(1,-1,1)$

$\boldsymbol{P}^2$	Irrep	$\boldsymbol{p}_2$	$\boldsymbol{p}_3$
4	A1	$(0,0,2)$	$(0,0,0)$
4	A1	$(0,0,1)$	$(0,0,1)$
4	A1	$(0,-1,1)$	$(0,1,1)$
4	B1	$(0,-1,1)$	$(0,1,1)$

And in the following table contains the same information for the three-pion sector.

$\boldsymbol{P}^2$	Irrep	$\boldsymbol{p}_1$	$\boldsymbol{p}_2$	$\boldsymbol{p}_3$
0	A1u	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$
0	A1u	$(0,0,-1)$	$(0,0,0)$	$(0,0,1)$
0	Eu	$(0,0,-1)$	$(0,0,0)$	$(0,0,1)$
1	A2	$(0,0,1)$	$(0,0,0)$	$(0,0,0)$
1	A2	$(-1,0,1)$	$(0,0,0)$	$(1,0,0)$
1	A2	$(0,0,-1)$	$(0,0,1)$	$(0,0,1)$
1	A2	$(-1,0,0)$	$(0,0,1)$	$(1,0,0)$
1	B2	$(-1,0,1)$	$(0,0,0)$	$(1,0,0)$
1	B2	$(-1,0,0)$	$(0,0,1)$	$(1,0,0)$
1	E	$(-1,0,1)$	$(0,0,0)$	$(1,0,0)$
2	A1	$(1,0,-1)$	$(0,0,0)$	$(0,1,1)$
2	A1	$(0,1,-1)$	$(0,0,1)$	$(1,0,0)$
2	A2	$(1,1,0)$	$(0,0,0)$	$(0,0,0)$
2	A2	$(1,1,-1)$	$(0,0,0)$	$(0,0,1)$
2	A2	$(1,0,-1)$	$(0,0,0)$	$(0,1,1)$
2	A2	$(0,1,0)$	$(0,0,0)$	$(1,0,0)$
2	A2	$(1,1,0)$	$(0,0,1)$	$(0,0,-1)$
2	A2	$(0,1,-1)$	$(0,0,1)$	$(1,0,0)$
2	A2	$(1,1,0)$	$(1,0,0)$	$(-1,0,0)$
2	A2	$(-1,1,0)$	$(1,0,0)$	$(1,0,0)$
2	B1	$(0,1,-1)$	$(0,0,1)$	$(1,0,0)$
2	B1	$(1,1,0)$	$(1,0,0)$	$(-1,0,0)$
2	B1	$(-1,1,0)$	$(1,0,0)$	$(1,0,0)$
2	B2	$(1,1,-1)$	$(0,0,0)$	$(0,0,1)$
2	B2	$(0,1,-1)$	$(0,0,1)$	$(1,0,0)$
3	A1	$(1,0,-1)$	$(0,0,1)$	$(0,1,1)$
3	A2	$(1,0,1)$	$(0,0,-1)$	$(0,1,1)$
3	A2	$(1,1,1)$	$(0,0,0)$	$(0,0,0)$

$\boldsymbol{P}^2$	Irrep	$\boldsymbol{p}_1$	$\boldsymbol{p}_2$	$\boldsymbol{p}_3$
3	A2	$(1,1,0)$	$(0,0,0)$	$(0,0,1)$
3	A2	$(1,1,1)$	$(0,0,1)$	$(0,0,-1)$
3	A2	$(1,1,-1)$	$(0,0,1)$	$(0,0,1)$
3	A2	$(1,0,-1)$	$(0,0,1)$	$(0,1,1)$
3	A2	$(0,1,0)$	$(0,0,1)$	$(1,0,0)$
3	E	$(1,0,1)$	$(0,0,-1)$	$(0,1,1)$
3	E	$(1,1,0)$	$(0,0,0)$	$(0,0,1)$
3	E	$(1,1,1)$	$(0,0,1)$	$(0,0,-1)$
3	E	$(1,1,-1)$	$(0,0,1)$	$(0,0,1)$
3	E	$(1,0,-1)$	$(0,0,1)$	$(0,1,1)$
3	E	$(0,1,0)$	$(0,0,1)$	$(1,0,0)$
4	A2	$(0,0,2)$	$(0,0,0)$	$(0,0,0)$
4	A2	$(0,0,1)$	$(0,0,0)$	$(0,0,1)$
4	A2	$(0,-1,1)$	$(0,0,0)$	$(0,1,1)$
4	A2	$(-1,0,1)$	$(0,0,1)$	$(1,0,0)$
4	B2	$(0,-1,1)$	$(0,0,0)$	$(0,1,1)$
4	B2	$(-1,0,1)$	$(0,0,1)$	$(1,0,0)$
4	E	$(-1,0,1)$	$(0,0,1)$	$(1,0,0)$

C. Acknowledgments

First I want to thank Carsten for the opportunity to write this thesis in his group. The past three years have been both very productive and pleasant. I have benefited much from discussions with all members of the group and beyond, getting help when I was stuck, giving pointers to better material, inspiration or just general motivation.

I thank Fernando for the most joyful collaboration on the energy spectrum analysis, which made us produce Reference [40].

Bartek and Matthias have helped to improve my work a lot by running supercomputer jobs and doubling the statistics on the physical point ensemble.

It has been a great pleasure to work with Markus on the ρ project. We both have learned a lot while designing and programming the analysis which culminated in Reference [30] and his PhD thesis. I was then able to use the whole toolchain to analyze the three pion interaction.

I'd like to thank Liuming for providing a cross-check of the three-pion contractions. This allowed to spot some issues early on before production.

The Collaborative Research Center 110 [104] has provided funding for my thesis position and covered travel expenses to Dublin (Ireland), Poznań (Poland), Wǔhàn and Běijīng (China). The Peking University has covered the costs of a three-week summer school. Also I got to attend the Lattice 2019 [105] conference to present Reference [30].

I was granted access to computer time on the JUWELS supercomputer [39] by the Joh von Neumann Institue for Computing (NIC) as well as general access to the QBIG cluster [106] of this work group. Without these resources this work would not have been possible.

Working on such a thesis certainly has its dreadful phases. I would like to thank Carsten, Bartek, other colleagues, my family, my partner and close friends for their support during these times and motivating me to keep going.

D. Curriculum Vitae

Full Name: Martin Hendrik Ueding

Date of Birth: 25th November 1990 in Bonn

Academic and school education:

- 10/2014 – 08/2017: **Master of Science in Phyics**, University of Bonn
- 10/2011 – 09/2014: **Bachelor of Science in Phyics**, University of Bonn
- 08/2001 – 06/2010: **Abitur**, Carl-von-Ossietzky-Gymnasium, Bonn

Bibliography

[1] L. Roper. "Evidence for a P-11 Pion-Nucleon Resonance at 556 MeV". In: *Phys. Rev. Lett.* 12 (1964), pp. 340–342. DOI: 10.1103/PhysRevLett.12.340.

[2] *Particle Data Group collaboration*: M. Tanabashi et al. "Review of Particle Physics". In: *Phys. Rev. D* 98.3 (2018), p. 030001. DOI: 10.1103/PhysRevD.98.030001.

[3] *Flavour Lattice Averaging Group collaboration*: S. Aoki et al. "FLAG Review 2019: Flavour Lattice Averaging Group (FLAG)". In: *Eur. Phys. J. C* 80.2 (2020), p. 113. DOI: 10.1140/epjc/s10052-019-7354-7. arXiv: 1902.08191 [hep-lat].

[4] M. E. Peskin and D. V. Schroeder. *An Introduction to Quantum Field Theory*. Westview Press, 1995. ISBN: 978-0-201-50397-5.

[5] M. Ueding. "Towards Exascale Computing for Lattice QCD". MA thesis. Aug. 2017. URL: https://martin-ueding.de/studies/msc_physics/physics900/index.html.

[6] H. Fritzsch, M. Gell-Mann, and H. Leutwyler. "Advantages of the Color Octet Gluon Picture". In: *Phys. Lett.* 47B (1973), pp. 365–368. DOI: 10.1016/0370-2693(73)90625-4.

[7] R. P. Feynman and A. R. Hibbs. *Quantum mechanics and path integrals*. International series in pure and applied physics. New York, NY: McGraw-Hill, 1965. URL: https://cds.cern.ch/record/100771.

[8] H. Rothe. *Lattice Gauge Theories*. Third Edition. Vol. 74. Singapore: World Scientific, 2005. ISBN: 981-256-168-4.

[9] S. Duane et al. "Hybrid Monte Carlo". In: *Phys. Lett.* B195 (1987), pp. 216–222. DOI: 10.1016/0370-2693(87)91197-X.

[10] T. Lippert. "The Hybrid Monte Carlo algorithm for quantum chromodynamics". In: *Lect. Notes Phys.* 508.122 (1998). DOI: 10.1007/BFb0106881. arXiv: hep-lat/9712019 [hep-lat].

[11] K. G. Wilson. "Confinement of quarks". In: *Phys. Rev. D* 10 (8 Oct. 1974), pp. 2445–2459. DOI: 10.1103/PhysRevD.10.2445. URL: https://link.aps.org/doi/10.1103/PhysRevD.10.2445.

[12] H. Nielsen and M. Ninomiya. "A no-go theorem for regularizing chiral fermions". In: *Physics Letters B* 105.2 (1981), pp. 219–223. ISSN: 0370-2693. DOI: 10.1016/0370-2693(81)91026-1.

[13] R. Frezzotti et al. "A Local formulation of lattice QCD without unphysical fermion zero modes". In: *Nucl. Phys. Proc. Suppl.* 83 (2000), pp. 941–946. DOI: 10.1016/S0920-5632(00)91852-8. arXiv: hep-lat/9909003 [hep-lat].

[14] R. Frezzotti. "Twisted mass lattice QCD". In: *Nucl. Phys. Proc. Suppl.* 140 (2005). [,134(2004)], pp. 134–140. DOI: 10.1016/j.nuclphysbps.2004.11.329. arXiv: hep-lat/0409138 [hep-lat].

[15] B. Sheikholeslami and R. Wohlert. "Improved continuum limit lattice action for QCD with wilson fermions". In: *Nuclear Physics B* 259.4 (1985), pp. 572–596. ISSN: 0550-3213. DOI: 10.1016/0550-3213(85)90002-1.

[16] *ETM collaboration*: A. Abdel-Rehim et al. "First physics results at the physical pion mass from $N_f = 2$ Wilson twisted mass fermions at maximal twist". In: *Phys. Rev.* D95.9 (2017), p. 094515. DOI: 10.1103/PhysRevD.95.094515. arXiv: 1507.05068 [hep-lat].

[17] L. Liu et al. "Isospin-0 $\pi\pi$ s-wave scattering length from twisted mass lattice QCD". In: *Phys. Rev.* D96.5 (2017), p. 054516. DOI: 10.1103/PhysRevD.96.054516. arXiv: 1612.02061 [hep-lat].

[18] L. Liu et al. "Isospin-0 $\pi\pi$ scattering from twisted mass lattice QCD". In: *PoS* LATTICE2016 (2017), p. 119. DOI: 10.22323/1.256.0119. arXiv: 1701.08961 [hep-lat].

[19] *ETM collaboration*: C. Helmes et al. "The η' meson at the physical point with $N_f = 2$ Wilson twisted mass fermions". In: *EPJ Web Conf.* 175 (2018). Ed. by M. Della Morte et al., p. 05025. DOI: 10.1051/epjconf/201817505025. arXiv: 1710.03698 [hep-lat].

[20] M. Fischer et al. “On the generalised eigenvalue method and its relation to Prony and generalised pencil of function methods”. In: (Apr. 2020). arXiv: `2004.10472 [hep-lat]`.

[21] M. Bhat et al. “Parton distribution functions from lattice QCD at physical quark masses via the pseudo-distribution approach”. In: (May 2020). arXiv: `2005.02102 [hep-lat]`.

[22] *ETM collaboration*: M. Fischer et al. “The ρ-resonance with physical pion mass from $N_f = 2$ lattice QCD”. In: (June 2020). arXiv: `2006.13805 [hep-lat]`.

[23] Y. Iwasaki. “Renormalization Group Analysis of Lattice Theories and Improved Lattice Action: Two-Dimensional Nonlinear O(N) Sigma Model”. In: *Nucl. Phys. B* 258 (1985), pp. 141–156. DOI: `10.1016/0550-3213(85)90606-6`.

[24] R. Baron et al. “Light hadrons from lattice QCD with light (u,d), strange and charm dynamical quarks”. In: *JHEP* 06 (2010), p. 111. DOI: `10.1007/JHEP06(2010)111`. arXiv: `1004.5284 [hep-lat]`.

[25] *ETM collaboration*: K. Ottnad and C. Urbach. “Flavor-singlet meson decay constants from $N_f = 2+1+1$ twisted mass lattice QCD”. In: *Phys. Rev. D* 97.5 (2018), p. 054508. DOI: `10.1103/PhysRevD.97.054508`. arXiv: `1710.07986 [hep-lat]`.

[26] *ETM collaboration*: C. Helmes et al. “Hadron-Hadron Interactions from $N_f = 2+1+1$ Lattice QCD: $I = 3/2$ πK Scattering Length”. In: *Phys. Rev. D* 98.11 (2018), p. 114511. DOI: `10.1103/PhysRevD.98.114511`. arXiv: `1809.08886 [hep-lat]`.

[27] C. Helmes et al. “Meson-meson scattering lengths at maximum isospin from lattice QCD”. In: *PoS* CD2018 (2019), p. 071. DOI: `10.22323/1.317.0071`. arXiv: `1904.00191 [hep-lat]`.

[28] C. Helmes. “K-K and π-K Scattering Lengths at Maximal Isospin from Lattice QCD”. PhD thesis. U. Bonn (main), 2019.

[29] F. Pittler. “Hadron-Hadron Interactions from $N_f = 2+1+1$ Lattice QCD: $\pi - K$ scattering length”. In: *PoS* LATTICE2018 (2019), p. 072. DOI: `10.22323/1.334.0072`.

[30] M. Werner et al. “Hadron-Hadron Interactions from $N_f = 2+1+1$ Lattice QCD: The ρ-resonance”. In: *European Physical Journal A* 56.2 (2019), p. 61. ISSN: 1434-601X. DOI: 10.1140/epja/s10050-020-00057-4. arXiv: 1907.01237 [hep-lat].

[31] M. Werner. “Hadron-Hadron Interactions from $N_f = 2+1+1$ Lattice QCD: The ρ-resonance”. PhD thesis. Bonn, Germany: Uni Bonn, 2019.

[32] B. Hörz and A. Hanlon. “Two- and three-pion finite-volume spectra at maximal isospin from lattice QCD”. In: *Phys. Rev. Lett.* 123.14 (2019), p. 142002. DOI: 10.1103/PhysRevLett.123.142002. arXiv: 1905.04277 [hep-lat].

[33] C. Lüdeling. *physics751: Group Theory (for Physicists)*. Aug. 2010. URL: http://www.th.physik.uni-bonn.de/nilles/people/luedeling/grouptheory.

[34] J. J. Dudek, R. G. Edwards, and C. E. Thomas. “S and D-wave phase shifts in isospin-2 pi pi scattering from lattice QCD”. In: *Phys. Rev.* D86 (2012), p. 034031. DOI: 10.1103/PhysRevD.86.034031. arXiv: 1203.6041 [hep-ph].

[35] M. Göckeler et al. “Scattering phases for meson and baryon resonances on general moving-frame lattices”. In: *Phys. Rev.* D86 (2012), p. 094513. DOI: 10.1103/PhysRevD.86.094513. arXiv: 1206.4141 [hep-lat].

[36] M. Ueding. *sLapH Projection Next Generation Code*. 2019. URL: https://github.com/HISKP-LQCD/sLapH-projection-NG.

[37] *Hadron Spectrum collaboration*: M. Peardon et al. “A Novel quark-field creation operator construction for hadronic physics in lattice QCD”. In: *Phys. Rev.* D80 (2009), p. 054506. DOI: 10.1103/PhysRevD.80.054506. arXiv: 0905.2160 [hep-lat].

[38] C. Morningstar et al. “Improved stochastic estimation of quark propagation with Laplacian Heaviside smearing in lattice QCD”. In: *Phys. Rev.* D83 (2011), p. 114505. DOI: 10.1103/PhysRevD.83.114505. arXiv: 1104.3870 [hep-lat].

[39] Jülich Supercomputing Centre. “JUWELS: Modular Tier-0/1 Supercomputer at the Jülich Supercomputing Centre”. In: *Journal of large-scale research facilities* 5.A135 (2019). DOI: 10.17815/jlsrf-5-171.

[40] M. Fischer et al. *Scattering of two and three physical pions at maximal isospin from lattice QCD*. Aug. 2020. arXiv: 2008.03035 [hep-lat].

[41] B. Blossier et al. *On the generalized eigenvalue method for energies and matrix elements in lattice field theory*. Feb. 2009. arXiv: 0902.1265 [hep-lat].

[42] M. Fischer. "Bayesian Inference in Analysing Results from Lattice QCD". MA thesis. Uni Bonn, May 2019.

[43] M. Ueding. *Fit Range Determination with Machine Learning*. 2020. URL: https://martin-ueding.de/posts/fit-range-determination-with-machine-learning.

[44] *Correlation, Variance and Covariance (Matrices)*. URL: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/cor.

[45] X. Feng, K. Jansen, and D. B. Renner. "The pi+ pi+ scattering length from maximally twisted mass lattice QCD". In: *Phys. Lett.* B684 (2010), pp. 268–274. DOI: 10.1016/j.physletb.2010.01.018. arXiv: 0909.3255 [hep-lat].

[46] C. Helmes et al. "Hadron-Hadron Interactions from $N_f = 2 + 1 + 1$ lattice QCD: Isospin-1 KK scattering length". In: *Phys. Rev. D* 96.3 (2017), p. 034510. DOI: 10.1103/PhysRevD.96.034510. arXiv: 1703.04737 [hep-lat].

[47] T. Umeda. "A Constant contribution in meson correlators at finite temperature". In: *Phys. Rev. D* 75 (2007), p. 094502. DOI: 10.1103/PhysRevD.75.094502. arXiv: hep-lat/0701005.

[48] A. Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. Paperback. O'Reilly Media, 2019. ISBN: 1492032646, 978-1492032649.

[49] M. Ueding. $N_\mathrm{f} = 2$ *three pion* $I = 3$ *scattering data repository*. URL: https://github.com/HISKP-LQCD/Nf2-3pi-I3-scattering-data.

[50] R. A. Briceno, J. J. Dudek, and R. D. Young. "Scattering processes and resonances from lattice QCD". In: *Rev. Mod. Phys.* 90.2 (2018), p. 025001. DOI: 10.1103/RevModPhys.90.025001. arXiv: 1706.06223 [hep-lat].

[51] R. A. Briceno et al. “Isoscalar $\pi\pi$ scattering and the σ meson resonance from QCD”. In: *Phys. Rev. Lett.* 118.2 (2017), p. 022002. DOI: 10.1103/PhysRevLett.118.022002. arXiv: 1607.05900 [hep-ph].

[52] Z. Fu. “Lattice QCD study of the s-wave $\pi\pi$ scattering lengths in the I=0 and 2 channels”. In: *Phys. Rev.* D87.7 (2013), p. 074501. DOI: 10.1103/PhysRevD.87.074501. arXiv: 1303.0517 [hep-lat].

[53] D. Guo et al. “Extraction of isoscalar $\pi\pi$ phase-shifts from lattice QCD”. In: *Phys. Rev.* D98.1 (2018), p. 014507. DOI: 10.1103/PhysRevD.98.014507. arXiv: 1803.02897 [hep-lat].

[54] C. Andersen et al. “The $I = 1$ pion-pion scattering amplitude and timelike pion form factor from $N_\mathrm{f} = 2 + 1$ lattice QCD”. In: *Nucl. Phys.* B939 (2019), pp. 145–173. DOI: 10.1016/j.nuclphysb.2018.12.018. arXiv: 1808.05007 [hep-lat].

[55] X. Feng, K. Jansen, and D. B. Renner. “Resonance Parameters of the rho-Meson from Lattice QCD”. In: *Phys. Rev.* D83 (2011), p. 094505. DOI: 10.1103/PhysRevD.83.094505. arXiv: 1011.5288 [hep-lat].

[56] Z. Fu and L. Wang. “Studying the ρ resonance parameters with staggered fermions”. In: *Phys. Rev.* D94.3 (2016), p. 034505. DOI: 10.1103/PhysRevD.94.034505. arXiv: 1608.07478 [hep-lat].

[57] L. Leskovec et al. “A Lattice QCD study of the ρ resonance”. In: *13th Conference on the Intersections of Particle and Nuclear Physics (CIPANP 2018) Palm Springs, California, USA, May 29-June 3, 2018*. 2018. arXiv: 1810.01927 [hep-lat].

[58] J. J. Dudek et al. “The phase-shift of isospin-2 pi-pi scattering from lattice QCD”. In: *Phys. Rev.* D83 (2011), p. 071504. DOI: 10.1103/PhysRevD.83.071504. arXiv: 1011.6352 [hep-ph].

[59] *ETM collaboration*: C. Helmes et al. “Hadron-hadron interactions from $\mathrm{N}_f = 2 + 1 + 1$ lattice QCD: isospin-2 $\pi\pi$ scattering length”. In: *JHEP* 09 (2015), p. 109. DOI: 10.1007/JHEP09(2015)109. arXiv: 1506.00408 [hep-lat].

[60] M. Mai et al. “Cross-channel study of pion scattering from lattice QCD”. In: *Phys. Rev. D* 100.11 (2019), p. 114514. DOI: 10.1103/PhysRevD.100.114514. arXiv: 1908.01847 [hep-lat].

[61] F. Schwabl. *Quantum mechanics*. 4th. Springer, 2007. ISBN: 9783540719328,3540719326.

[62] T. D. Blanton, F. Romero-López, and S. R. Sharpe. “$I = 3$ three-pion scattering amplitude from lattice QCD”. In: *Phys. Rev. Lett.* 124.3 (2020), p. 032001. DOI: 10.1103/PhysRevLett.124.032001. arXiv: 1909.02973 [hep-lat].

[63] F. Yndurain. “Low-energy pion physics”. In: (Dec. 2002). arXiv: hep-ph/0212282.

[64] J. Pelaez and F. Yndurain. “The Pion-pion scattering amplitude”. In: *Phys. Rev. D* 71 (2005), p. 074016. DOI: 10.1103/PhysRevD.71.074016. arXiv: hep-ph/0411334.

[65] R. Kaminski, J. Pelaez, and F. Yndurain. “The Pion-pion scattering amplitude. III. Improving the analysis with forward dispersion relations and Roy equations”. In: *Phys. Rev. D* 77 (2008), p. 054015. DOI: 10.1103/PhysRevD.77.054015. arXiv: 0710.1150 [hep-ph].

[66] M. Lüscher. “Volume Dependence of the Energy Spectum in Massive Quantum Field Theories”. In: *Commun. Math. Phys.* 105 (1986), pp. 153–188.

[67] M. Petschlies and C. Urbach. *rzeta*. URL: https://github.com/HISKP-LQCD/rzeta.

[68] G. Colangelo, J. Gasser, and H. Leutwyler. “$\pi\pi$ scattering”. In: *Nucl. Phys. B* 603 (2001), pp. 125–179. DOI: 10.1016/S0550-3213(01)00147-X. arXiv: hep-ph/0103088.

[69] M. T. Hansen and S. R. Sharpe. “Lattice QCD and Three-particle Decays of Resonances”. In: *Ann. Rev. Nucl. Part. Sci.* 69 (2019), pp. 65–107. DOI: 10.1146/annurev-nucl-101918-023723. arXiv: 1901.00483 [hep-lat].

[70] S. R. Beane et al. “Multi-Pion Systems in Lattice QCD and the Three-Pion Interaction”. In: *Phys. Rev. Lett.* 100 (2008), p. 082004. DOI: 10.1103/PhysRevLett.100.082004. arXiv: 0710.1827 [hep-lat].

[71] S. R. Beane et al. “Precise Determination of the I=2 pi pi Scattering Length from Mixed-Action Lattice QCD”. In: *Phys. Rev. D* 77 (2008), p. 014505. DOI: 10.1103/PhysRevD.77.014505. arXiv: 0706.3026 [hep-lat].

[72] W. Detmold et al. “Multi-Pion States in Lattice QCD and the Charged-Pion Condensate”. In: *Phys. Rev. D* 78 (2008), p. 014507. DOI: 10.1103/PhysRevD.78.014507. arXiv: 0803.2728 [hep-lat].

[73] F. Romero-López, A. Rusetsky, and C. Urbach. “Two- and three-body interactions in φ^4 theory from lattice simulations”. In: *Eur. Phys. J.* C78.10 (2018), p. 846. DOI: 10.1140/epjc/s10052-018-6325-8. arXiv: 1806.02367 [hep-lat].

[74] M. Mai and M. Doring. “Finite-Volume Spectrum of $\pi^+\pi^+$ and $\pi^+\pi^+\pi^+$ Systems”. In: *Phys. Rev. Lett.* 122.6 (2019), p. 062503. DOI: 10.1103/PhysRevLett.122.062503. arXiv: 1807.04746 [hep-lat].

[75] T. D. Blanton, F. Romero-López, and S. R. Sharpe. “Implementing the three-particle quantization condition including higher partial waves”. In: *JHEP* 03 (2019), p. 106. DOI: 10.1007/JHEP03(2019)106. arXiv: 1901.07095 [hep-lat].

[76] M. Mai et al. “Three-body unitarity versus finite-volume $\pi^+\pi^+\pi^+$ spectrum from lattice QCD”. In: *Phys. Rev.* D101.5 (2020), p. 054510. DOI: 10.1103/PhysRevD.101.054510. arXiv: 1909.05749 [hep-lat].

[77] M. T. Hansen, F. Romero-López, and S. R. Sharpe. “Generalizing the relativistic quantization condition to include all three-pion isospin channels”. In: (2020). arXiv: 2003.10974 [hep-lat].

[78] S. Beane et al. “Charged multi-hadron systems in lattice QCD+QED”. In: (Mar. 2020). arXiv: 2003.12130 [hep-lat].

[79] C. Culver et al. “Three body spectrum from lattice QCD”. In: (2019). arXiv: 1911.09047 [hep-lat].

[80] C. Urbach, B. Kostrzewa, and S. Reker. *hadron Repository*. URL: https://github.com/etmc/hadron.

[81] H. Wickham et al. “Tidy data”. In: *Journal of Statistical Software* 59.10 (2014), pp. 1–23.

[82] M. Ueding. *Parameter-Value-Framework*. 2018. URL: https://github.com/HISKP-LQCD/paramvalf.

[83] H. Wickham et al. *dplyr: A Grammar of Data Manipulation*. R package version 0.8.3. 2019. URL: https://CRAN.R-project.org/package=dplyr.

[84] H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN: 978-3-319-24277-4. URL: `https://ggplot2.tidyverse.org`.

[85] J. Ellson et al. URL: `https://www.graphviz.org`.

[86] H. Wickham. *Advanced R, Second Edition (Chapman & Hall/CRC The R Series)*. Chapman and Hall/CRC, June 2019. ISBN: 0815384572. URL: `https://www.xarg.org/ref/a/0815384572/`.

[87] K. Kuang, Q. Kong, and F. Napolitano. *pbmcapply: Tracking the Progress of Mc*pply with Progress Bar*. R package version 1.5.0. 2019. URL: `https://CRAN.R-project.org/package=pbmcapply`.

[88] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2020. URL: `https://www.R-project.org/`.

[89] M. Werner. *sLapH Projection Code*. URL: `https://github.com/HISKP-LQCD/sLapH-projection`.

[90] Wolfram Research Inc. *Wolfram Language*. URL: `https://www.wolfram.com/language`.

[91] D. Djukanovic. "Quark Contraction Tool – QCT". In: (2016). arXiv: 1603.01576 [hep-lat].

[92] B. Knippschild, M. Ueding, and M. Werner. *sLapH Contractions Repository*. URL: `https://github.com/maowerner/sLapH-contractions`.

[93] *Travis CI*. URL: `https://travis-ci.org`.

[94] Wikipedia contributors. *Correlation does not imply causation — Wikipedia, The Free Encyclopedia*. 2020. URL: `https://en.wikipedia.org/w/index.php?title=Correlation_does_not_imply_causation&oldid=935982928`.

[95] T. Vigen. *Spurious Correlations*. URL: `https://www.tylervigen.com/spurious-correlations`.

[96] I. P. Omelyan, I. M. Mryglod, and R. Folk. "New optimized algorithms for molecular dynamics simulations". In: *Condensed Matter Physics* 5.3(31) (2002), pp. 369–390. URL: `http://www.icmp.lviv.ua/journal/zbirnyk.31/001/art01.pdf`.

[97] E. Jones, T. Oliphant, P. Peterson, et al. *SciPy: Open source scientific tools for Python*. 2001–. URL: http://www.scipy.org.

[98] D. Maclaurin et al. *Autograd*. URL: https://github.com/HIPS/autograd.

[99] *YAML Ain't Markup Language*. URL: https://yaml.org.

[100] U. Noè and M. Ueding. *DAG MCMC*. URL: https://github.com/unoe/dag-mcmc.

[101] M. Davis. *SnakeViz: An in-browser Python profile viewer*. URL: https://jiffyclub.github.io/snakeviz.

[102] S. K. Lam, A. Pitrou, and S. Seibert. "Numba: A LLVM-Based Python JIT Compiler". In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. LLVM '15. Austin, Texas: Association for Computing Machinery, 2015. ISBN: 9781450340052. DOI: 10.1145/2833157.2833162. URL: https://doi.org/10.1145/2833157.2833162.

[103] J. Bradbury et al. *JAX: composable transformations of Python+NumPy programs*. URL: http://github.com/google/jax.

[104] *CRC 110*. URL: https://crc110.hiskp.uni-bonn.de.

[105] *The 37th Annual International Symposium on Lattice Field Theory*. URL: http://lattice2019.ccnu.edu.cn.

[106] C. Urbach. *QBiG Cluster*. URL: https://www.itkp.uni-bonn.de/~urbach/gpucluster.html.

[107] D. E. Knuth, T. Larrabee, and P. M. Roberts. *Mathematical Writing*. 1987. URL: http://jmlr.csail.mit.edu/reviewing-papers/knuth_mathematical_writing.pdf.