The pion Form Factor from Lattice QCD
van der Heide, J.

Citation for published version (APA):
van der Heide, J. (2004). The pion Form Factor from Lattice QCD

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Chapter 3.

Simulation techniques

In this chapter the structure of the simulation program will be outlined, and calculational tools and the choices for parameters are discussed. We start with the generation of configurations of gauge links. The construction of $n$-point Green’s functions from elementary quark propagators is discussed together with the discrete symmetries which enable us to rewrite the Wick contracted correlation functions. We then outline some additional numerical concepts motivated by physics arguments, followed by the numerical inversion of the fermion matrix. We end the chapter with an overview of the simulation parameters used in this work.

3.1. Gauge section

After the analytic integration over the fermionic degrees of freedom, using the quenched approximation, the path integral has the following form

$$\langle O(U) \rangle = \frac{1}{Z} \int \mathcal{D}U e^{-S(U)} O(U).$$  \hspace{1cm} (3.1)

Since this integral has a very large number of degrees of freedom, which scales with the number of lattice points, usual numerical integration methods cannot be applied. Instead, one has to turn to so-called Monte Carlo techniques. The straightforward application of this method would consist of picking the set of gauge fields randomly from a uniform distribution and evaluate Eq. 3.1 on each of them. Since such a sample is necessarily finite, Eq. 3.1 changes

$$\langle O(U) \rangle \approx \frac{1}{Z} \sum_i e^{-S(U_i)} O(U_i).$$  \hspace{1cm} (3.2)

Now, all fields are equally probable but their contribution to the integral might be insignificant. This means that one would have to ‘walk’ through the complete gauge configuration domain in order to have a sensible estimate of the integral. This in turn leads to a very slow convergence. This is most severe when the integrand varies rapidly or the dimensionality of the integral becomes large. Our integral has both these properties and we therefore need a different scheme.

Instead, we turn to importance sampling, which amounts to generating a set of representative gauge field configurations on which the observables are calculated. The
generation of a suitable set is a highly non-trivial task. It took some time to find an algorithm that included both an acceptable convergence rate and autocorrelation time. This will be the subject of the next section.

3.1.1. Markov chains and Metropolis

To generate configurations that have a significant contribution to the integral (without ruling out the others completely), we use importance sampling, i.e. we pick the configurations with a probability distribution of the form

\[ P(U) = \frac{1}{Z} e^{-S_G(U)}. \]  

(3.3)

Since we are studying equilibrium physics, a useful method to generate a sample of configurations with the probability distribution \( P(U) \) is the use of a so-called Markov process. The idea is to generate a new configuration \( U' \) from an old one with a transition probability \( P_T(U, U') \). This update is then repeated a number of times to create a chain of configurations that, after some thermalisation is distributed according to Eq. 3.3. The algorithm must have several properties to ensure that this is the case. First of all the probability of reaching any configuration \( U' \) from any other must be nonzero

\[ P_T(U, U') > 0 \quad \text{for all } U \text{ and } U'. \]  

(3.4)

This is called ergodicity. Second, the transition must preserve the probability distribution Eq. 3.3, i.e

\[ P(U') = \int dU P(U) P_T(U, U') \quad \text{for all } U'. \]  

(3.5)

A necessary condition, ensuring that the chain of configurations has the appropriate distribution irrespective of the starting configuration is detailed balance,

\[ P(U') P_T(U', U) = P(U) P_T(U, U') \quad \text{for all } U'. \]  

(3.6)

Metropolis et al. [41] introduced a recipe for the generation of a Markov chain by picking the trial link \( U' \) randomly and using the acceptance probability

\[ P_A(U, U') = \min \left( 1, \frac{P(U')}{P(U)} \right). \]  

(3.7)

Together, they form the transition function, which ensures that the chain satisfies detailed balance and ergodicity. The update is normally done on a single link \( U \), since changing all the links before applying the acceptance criterion drives the acceptance rate to zero. Updating a configuration \( (U) \) thus consists of updating all individual links \( (U) \) separately.
The main problem of the Metropolis algorithm is that it either has a low acceptance rate or the correlation time is long because of high correlations between successive configurations. The autocorrelation time $\tau$ depends on the lattice correlation length $\xi$ as

$$\tau \sim \xi^z,$$

with $z$ the dynamical critical exponent. For the Metropolis algorithm $z \approx 2$ and thus $\tau$ can become large, especially in the vicinity of the continuum limit where the correlation length diverges. This problem will be addressed in the next section, where several improvements of the above method are discussed.

### 3.1.2. Heat bath and overrelaxation

The Wilson action has the important advantage that it is local (nearest neighbour interactions only). The change in the action due to updating one link can thus be calculated very fast

$$S_G = R \text{Tr} U V + \text{terms not involving } U,$$

with $V$ the sum of the product of remaining link variables of the plaquettes containing $U$. It is therefore feasible to update a single link a number of times and then proceed to the next link. This is called multi-hit Metropolis. For a large number of updates per link, this procedure is equivalent to the heat bath algorithm [42]. The heat bath algorithm updates an individual link using the differential probability distribution determined by the neighbouring links

$$dP(U) = e^{R \text{Tr} U V} dU,$$

Stated differently, it is set into local equilibrium with them. The heat bath algorithm was originally proposed for $SU(2)$, but applied to $SU(3)$ [43] it was rather slow. Cabibbo and Marinari [44] therefore suggested to use it to update the $SU(2)$ subgroups of the link. This work was further improved and is known as the FHKP updating scheme [45, 46]. It will be used in this work. This crucial difference between the heat bath method and the Metropolis algorithm (which updates a link based on the old link itself) is that the new link is much less correlated with the old one. One heat bath update step then consists of 'sweeping' through the complete lattice. It reduces $\xi$, but the dynamical critical exponent, however is still $z \approx 2$.

With the overrelaxation algorithm [47, 48] one can reduce the correlation between two consecutive configurations even more. Working again in the subgroups of the $SU(3)$ links, one chooses a $SU(2)$ matrix lying opposite in parameter space to the original one without changing the action $S_G(U)$. Since the change in the action is zero, the transition is always accepted. The method is obviously not ergodic, so one can only use it in combination with another algorithm which supplies the ergodicity. Depending on the number of overrelaxation steps per heat bath update, it is possible to lower the critical exponent to $z \approx 1$. 

---

25
Chapter 3. Simulation techniques

For our choice of the coupling, we will use the combination of one heat bath sweep and four overrelaxation steps.

### 3.2. The quark propagator and pion \( n \)-point Green’s functions

Since all the pion Green’s functions used in this work are calculated using quark propagators, it is useful to recapitulate their basic properties and implications on the pion \( n \)-point Green’s functions [10]. Using charge conjugation, with the property

\[
C \gamma_\mu C^{-1} = -\gamma_\mu^T, \quad C \gamma_5 C^{-1} = \gamma_5^T, \tag{3.11}
\]

we find for the inverse the quark propagator\(^1\)

\[
CM_f^{-1}(x, y; \{U\})C^{-1} = M_f^{-1}(y, x; \{U^*\})^T. \tag{3.12}
\]

where the transpose is over the colour and Dirac indices and \( f \) denotes the flavour. \( \{U\} \) denotes the gauge field configuration on which the propagator has been calculated. Since the gauge action is invariant under the transformation \( U \to U^* \) we find \( \exp(-S_G(\{U^*\})) = \exp(-S_G(\{U\})) \). Because of Eq. 3.12, a general pion \( n \)-point Green’s function can thus be written as

\[
\langle G(\{U\}) \rangle_\mathcal{U} = \frac{1}{2} \langle G(\{U\}) + G(\{U^*\}) \rangle_\mathcal{U}. \tag{3.13}
\]

The subscript \( \mathcal{U} \) denotes the ensemble average over the gauge configurations (cf. Sec. 3.1) Another useful operator is \( \tilde{C} = C \gamma_5 \). It has the property

\[
\tilde{C} \gamma_\mu \tilde{C}^{-1} = \gamma_\mu^T, \quad \tilde{C} \gamma_5 \tilde{C}^{-1} = \gamma_5^T, \tag{3.14}
\]

and results in the following identity

\[
\tilde{C}M_f^{-1}(x, y; \{U\})\tilde{C}^{-1} = \left( M_f^{-1} \right)^*(x, y; \{U^*\}). \tag{3.15}
\]

Using Eqs. 3.12 and 3.15 one finds the \( \gamma_5 \)-symmetry

\[
\gamma_5 M_f^{-1}(x, y; \{U\}) \gamma_5 = \left( M^{-1} \right)^+_f (y, x; \{U\}) \tag{3.16}
\]

With these two equations it is easy to show that for our Green’s functions,

\[
G(\{U^*\}) = G^*(\{U\}) \tag{3.17}
\]

---

\(^1\)Remember that \( M_f^{-1}(x, y; \{U\}) = G_f(x, y; \{U\}) \). To avoid possible confusion with the pion Green’s functions, we choose to represent the propagator by the inverse of the fermion matrix instead.
and thus
\[ \frac{1}{2} \left( G(\{U\}) + G(\{U^*\}) \right) = \Re(G(\{U\})) . \]  

(3.18)

This shows that we can use the real part as our signal, and discard the imaginary part, which averages out to zero for an infinite sample. Using parity, one can show that the pion n-point correlation functions are even under spatial inversion and therefore the Green’s function in momentum space should be real. In the next sections we will specialise to two- and three-point Green’s functions.

### 3.3. Pion two-point Green’s function

Having discussed general n-point Green’s functions in the previous section, we now specialise to the two point function for the pion, given by

\[ G(x, x_i) = \langle \Omega | \phi(x) \phi_i^\dagger(x_i) | \Omega \rangle . \]  

(3.19)

Using the creation/annihilation operators for the pion, restricting ourselves to a \( \pi^+ \),
\[ \phi(x) = \bar{\psi}_d(x) \gamma^5 \psi_u(x) \quad \quad \phi_i^\dagger(x) = -\bar{\psi}_u(x) \gamma^5 \psi_d(x), \]  

(3.20)

and dropping the hats on dimensionless fields from now on, we can rewrite Eq. 3.19 using Wick’s theorem to obtain

\[ G(x, x_i) = \langle \text{Tr} \left( \gamma^5 M_d^{-1}(x, x) \gamma^5 M_u^{-1}(x, x_i) \right) \rangle_U . \]  

(3.21)

Here \( M_u^{-1}(x, x_i) \) represents the propagation of a \( u \) quark from \( x_i \) to \( x \) and we have dropped the dependence on \( U \). A similar interpretation holds for \( M_d^{-1}(x_i, x) \). The subscript \( U \) denotes the configuration average, i.e. the expectation value with respect to the sets of gauge links as explained in Sec. 3.1. In order to calculate the pion Green’s function one thus needs two quark propagators. In Lattice QCD, where the inversion of the fermion matrix is the crucial and most time consuming step in the complete calculation, this is not desirable. But as we have seen in Eq. 3.16, the forward and backward propagators are related due to the discrete symmetries on the lattice. Rewriting the 'backward' propagator, we then find for our two-point function \[49]\]

\[ G(x, x_i) = \langle \text{Tr} \left( (M^{-1})_d^\dagger(x, x_i) M_u^{-1}(x, x_i) \right) \rangle_U , \]  

(3.22)

which means that for degenerate flavours \( u \) and \( d \), we need to invert (part of) the matrix only once. In momentum space the two-point Green’s function can be represented as

\[ \tilde{G}(t, t_i; p) = \sum_x e^{-i p \cdot (x - x_i)} G(x, x_i) . \]  

(3.23)

Substituting Eq. 3.22 in Eq. 3.23 and making use of Eqs. 3.13 and 3.18 one finally obtains

\[ \tilde{G}(t, t_i; p) = \langle \text{Tr} \Re \sum_x e^{-i p \cdot (x - x_i)} M_d^{-1}(x, x_i) M_u^{-1}(x, x_i) \rangle_U . \]  

(3.24)
In this form the two-point function is easily implemented in a computer program. The discussion regarding the inversion of the Dirac matrix will be postponed to after the discussion of the three point function.

The two-point function contains information about energies and amplitudes of the meson under consideration. These observables, including the dispersion relation, have all been calculated before by others and can therefore serve as independent checks on our methods. Furthermore, the parametrisation of the 3-point function contains several of these parameters. To extract the form factor, it is therefore necessary to obtain the energies and amplitudes from another observable, the two-point function.

### 3.3.1. Operator and gauge field smearing

The 'pion operator' introduced in this Sec. 3.3 projects not only on the ground state of the pion, but also onto the other states having the same quantum numbers. In practise, however, these will be the first excited states of the pion: as we will see later on, the two-point correlation function is exponentially damped with the energy times the propagation time of the pions. Measuring the correlation function for increasing separation time will thus provide a means to remove the excited states one after the other and eventually filter out the ground state. In principle this would work for a large enough temporal extension of the lattice. In our case, it is certainly possible to obtain the masses and energies in this way, even though we can only use half of the lattice because of 'wrap around' effects due to our periodic boundary conditions. However, for the three-point function a problem arises since in this case one is only able to use approximately one third of the extent of the lattice because of other, more severe wrapping around effects (see Sec. 5.2). It is however possible to improve the 'pion operators' to enhance the overlap with the ground state pion. The first observation to make is that the pion operators are not physical in the sense that they lack spatial extension. Thus a promising improvement step is to use non-local pion operators, i.e. spatially separated quark and anti-quark operators. To preserve gauge invariance, one connects the (separated) quarks by gauge links. The physical interpretation of this connection is of course the gluonic tube, giving rise to the potential energy of the pair. There are several schemes to implement this operator smearing in a gauge invariant way [50,51]. In this work the scheme developed by Gupta et al. [52] and Lacock et al. [53] is used. The quarks are separated by a distance $R$, and connected with a linear string of gauge links. This is done for every spatial direction (see Fig. 3.1(a)) in order to build an extended operator which produces an S-wave in accordance with the quantum numbers of the ground state of the meson. The smeared operator, which we only use on the sink side, can be written as

$$\phi_R(x) = \tilde{\psi}_d(x) \gamma_5 \psi_u(x),$$  \hspace{1cm} (3.25)
3.3. Pion two-point Green’s function

where

\[
\tilde{\psi}_u(x) = \frac{1}{6} \sum_{\mu=1}^{3} (U^\dagger_\mu(x - \hat{\mu})...U^\dagger_\mu(x - R\hat{\mu})\psi_u(x - R\hat{\mu}) + U_\mu(x)...U_\mu(x + (R - 1)\hat{\mu})\psi_u(x + R\hat{\mu})) .
\]  

(3.26)

The smearing is thus performed on the \( u \)-propagator where after it is connected to the \( d \)-propagator. In this way it is possible to perform the calculation of different \( R \)-values iteratively and thus save computer time. The optimal smearing must be determined from simulation by inspection of \textit{e.g.} the effective energy (See Sec. 4.3).

The reason to only smear the sink side is simply a computational one. We create fermion propagators originating from one point \((0,0)\). If we would want to have a non-local source operator, we should also calculate the propagators coming from many other points close to the origin. The huge amount of extra computing time makes this unfeasible.

![Diagram](a) + \epsilon \{ \text{Staples} \}

**Figure 3.1:** Overlap enhancement. Operator- (a) and gauge field smearing (b)

In order to further enhance the overlap, one can try to better mimic the gluonic tube. The enhancement is likely to be much better if the linear string of links is somewhat spread in space. The \textit{APE-smearing technique} [54] is such an improvement. It consists of adding to a link the staples multiplied with a weight factor \(\epsilon\) and projecting back to \(SU(3)\). The staples are built from the link variables that would complete a plaquette. It is done for the spatial directions perpendicular to the link which is being smeared (See Fig. 3.1(b)). This procedure should be repeated \(n\) times. Studies have shown [53] that the optimal overlap is achieved for small \(\epsilon\) and large \(n\). However, the dependence on the exact values of these parameters is very small, as long as one \textit{uses} gauge smearing. In order to save computer time, we therefore choose \(\epsilon = 0.5\) and the number of iterations \(n = 4\).

A byproduct of operator smearing is that one is able to obtain the 'Bethe-Salpeter amplitudes' from the two-point function. This will be dealt with in Sec. 4.5.
Chapter 3. Simulation techniques

3.4. Vector mesons

In the discussion of the form factor, we will often refer to the vector meson dominance model. Therefore it is natural to also obtain the mass of the lightest vector meson, i.e. the $\rho$-meson, consistently with our methods. The creation/annihilation operators of Eq. 3.20 are now different

\[ V^k(x) = \bar{\psi}_d(x) \gamma^k \psi_u(x) \quad V^{k\dagger}(x) = -\bar{\psi}_u(x) \gamma^k \psi_d(x), \quad (3.27) \]

Eq. 3.24 changes to

\[ \tilde{G}_\rho^k(t, t_i; p) = \langle \text{Tr} \, \tilde{R} \sum_x e^{-ip(x-x_i)} \gamma^k \gamma^5 M_d^{-\dagger}(x, x_i; \{U\}) \gamma^5 \gamma^k M_u^{-1}(x, x_i; \{U\}) \rangle_U. \]

(3.28)

In the calculation, the polarisation states perpendicular to the correlation direction, denoted by $k$, are averaged over.

3.5. The three-point Green’s function

The three-point function contains information on the form factor and is therefore of central importance. In the space-time representation it reads

\[ G_\mu(x_f, x, x_i) = \langle \Omega | T \left( \phi_\mu(x_f) j_\mu(x) \phi^\dagger_n(x_i) \right) | \Omega \rangle. \]

(3.29)

On the lattice one has several choices for the current $j_\mu$. The local current, corresponding to the continuum one, reads

\[ j^L_\mu(x) = 2 q_f \kappa_f \bar{\psi}_f(x) \gamma_\mu \psi_f(x). \]

(3.30)

Here, $q_f$ denotes the quark charge and the subscript $f$ can be either $u$ or $d$ (a summation is assumed), depending on which quark is ‘probed’ with the current. The local current is however not conserved on the lattice and needs renormalisation, leading to the renormalised local current

\[ j^{RL}_\mu(x) = Z_V j^L_\mu \]

(3.31)

with

\[ Z_V = Z^0_V (1 + b_V m_q) \]

(3.32)

the (mass dependent) renormalisation constant. The quark mass $m_q$, is defined in Eq. 2.21. Despite renormalisation, this current is still only conserved to $O(a)$.

Applying the Noether procedure, one can construct a conserved current [23]

\[ j^G_\mu(x) = q_f \kappa_f \left[ \bar{\psi}_f(x) (1 - \gamma_\mu) U_\mu(x) \psi_f(x + \mu) - \bar{\psi}_f(x + \mu) (1 + \gamma_\mu) U^{\dagger}_\mu(x) \psi_f(x) \right]. \]

(3.33)

This current is conserved but receives $O(a)$ corrections away from the forward direction. As mentioned in Sec. 2.3.2, in order to obtain full $O(a)$-improvement, one also needs to
3.5. The three-point Green’s function

improve local composite operators. Using Symanzik’s improvement program [34], one can identify the appropriate operators, which, when used together with the improved action, result in matrix elements that are free of all \( \mathcal{O}(a) \) discretisation errors. The operator which in our case fulfils this requirement is the tensor term

\[
j^T_\mu(x) = 2q_f \kappa_f \partial_\nu T_{\mu\nu}
\]

with

\[
T_{\mu\nu} = i \bar{\psi}_f(x) \sigma_{\mu\nu} \psi_f(x).
\]

The improved current \([55,56]\) can then easily be constructed from the local one as

\[
j^I_\mu = Z_V \left\{ j^L_\mu + c_V j^T_\mu \right\},
\]

where \( c_V \) denotes the improvement parameter. Since the improved current is a linear combination of the local current and the tensor term, we calculate their expectation values separately. In the analysis the two are combined to form the improved current. In this way it also remains possible to re-analyse the data with a different value for \( c_V \). There exists also a conserved improved current. It is based on the conserved current, with the same tensor operator for the \( \mathcal{O}(a) \) correction, but with a different improvement constant \( c'_V \). At the time of this study, this constant had not yet been determined non-perturbatively, thus we only use the improved current.

For convenience, we write the different currents more compactly as

\[
j^{\{L,C,I\}}_\mu(x) = q_f \kappa_f \sum_{y,y'} \bar{\psi}_f(y) K^{\{L,C,I\}}_\mu(x,y,y') \psi_f(y')
\]

with

\[
K^L(x,y,y') = 2\gamma_\mu \delta_{x,y} \delta_{x,y'}
\]

\[
K^C(x,y,y') = (1 - \gamma_\mu) U_\mu(x) \delta_{x,y} \delta_{x+\bar{\nu},y} - (1 + \gamma_\mu) U^\dagger_\mu(x) \delta_{x+\bar{\mu},y} \delta_{x,y'}
\]

\[
K^I(x,y,y') = 2\gamma_\mu \delta_{x,y} \delta_{x,y'} + 2i c_V (\delta_{x+\bar{\nu},y} \sigma_{\mu\nu} \delta_{x+y'} - \delta_{x-\bar{\nu},y} \sigma_{\mu\nu} \delta_{x+y'})
\]

Following the same steps as in Sec. 3.3, one sees that the three point function splits into a connected- and a disconnected part, Fig 3.2

\[
G_\mu(x_f,x,x_i) = \langle G^{\text{disc}}_\mu(x_f,x,x_i;\{U\}) + G^{\text{conn}}_\mu(x_f,x,x_i;\{U\}) \rangle_U,
\]

where the two parts can be written as

\[
G^{\text{disc}}_\mu(\{U\}) = \text{Tr} \left[ M^{-1}_u(x_f,x_i) \gamma^5 M^{-1}_d(x_i,x_f) \gamma^5 \right] \times
\]

\[
\sum_{f=u,d} q_f \kappa_f \sum_{y,y'} \text{Tr} \left( K^{\mu}(x,y,y') M^{-1}_f(y,y') \right)
\]

31
Performing analogous steps as in Sec. 3.2, one obtains

\[ CK^\mu(x, y, y'; \{U^*\}) C^{-1} = -K^T_\mu(x, y, y'; \{U^*\}). \]

(3.42)

For the disconnected part we thus find

\[ G^\mu_{\text{disc}}(\{U^*\}) = -G^\mu_{\text{disc}}(\{U\}). \]

(3.43)

Therefore, upon using Eq. 3.13, we find that the disconnected diagrams do not contribute [10]. Similarly, the connected part can be written as

\[ G^\mu_{\text{conn}}(\{U\}) = \sum_{y, y'} \text{Tr} \left\{ -q_u \kappa_u M_{ud}^{-1}(x_f, x_i; \{U^*\}) \gamma^5 M_{ud}^{-1}(x_i, y; \{U^*\}) \times \right. \]

\[ K_\mu(x, y; \{U^*\}) M_{ud}^{-1}(y', x_f; \{U^*\}) \gamma^5 + q_d \kappa_d M_{ud}^{-1}(x_f, x_i; \{U\}) \times \]

\[ \gamma^5 M_{ud}^{-1}(x_i, y; \{U\}) K_\mu(x, y, y'; \{U\}) M_{ud}^{-1}(y', x_f; \{U\}) \gamma^5 \right\} \]

(3.44)

Since \( U \) and \( U^* \) are equally probable in the ensemble average, Eq. 3.44 simplifies to

\[ G^\mu_{\text{conn}}(\{U\}) = \kappa \sum_{y, y'} \text{Tr} \left\{ M^{-1}(x_f, x_i) \gamma^5 M^{-1}(x_i, y) K_\mu(x, y, y') M^{-1}(y', x_f) \gamma^5 \right\} \]

(3.45)

where we assumed degenerate quarks (\( \kappa_u = \kappa_d = \kappa \)), dropped any reference to flavour and do not show the pion charge, \( q_\pi = q_u - q_d = 1 \).
3.6. Inversion of the fermion matrix

The three-momentum representation of the three point function is given by

$$\tilde{G}_\mu(t_f, t_i; p_f, q) = \sum_{x_f} \sum_x e^{-ip \cdot (x_f - x) + iq \cdot (x - x_i)} G_\mu(x_f, x, x_i).$$  (3.46)

Substituting Eqs. 3.39 and 3.45 into 3.46 and using Eqs. 3.16, 3.18, and the fact that the disconnected part does not contribute, we find for our three point function in momentum space

$$\tilde{G}(t_f, t_i; p_f, q) = \kappa \langle \text{Tr} \sum_{y, y'} M^{-1}(y, x_i; \{U\}) \gamma^5 \sum_x K_\mu(x, y, y'; \{U\}) e^{iq \cdot x}$$
$$\times \sum_{x_f} M^{-1}(x_f, x_i; \{U\}) \gamma^5 M^{-1}(x_f, x_i; \{U\}) e^{-ip \cdot x_i} U \rangle.$$  (3.47)

In the next section, we will discuss how to obtain the quark propagators and in particular how to obtain the second line in Eq. 3.47.

3.6. Inversion of the fermion matrix

Our pion Green's functions can be seen to consist of combinations of quark propagators as the building blocks. In order to obtain these propagators, it is necessary to invert the fermion matrix. The inverse of the matrix can be obtained in a reasonable amount of time since our action has nearest-neighbour interactions only. This means that the matrix is nearly diagonal. Would the action be less local, the inversion would probably take more time, and the memory of one processor would not suffice to contain the complete matrix. As can be done for any invertible matrix, we write the inversion as a set of linear equations which can be solved one at a time,

$$\sum_y M_{\alpha\beta}^a(x, y) V^b_\beta(y) = \eta^a_\alpha(x).$$  (3.48)

This equation can be solved numerically by choosing the appropriate source term on the right-hand side. The general solution of the above equation reads

$$V^c_{\gamma}(z) = \sum_x (M^{-1})^{ca}_{\gamma\alpha}(z, x) \eta^a_\alpha(x).$$  (3.49)

Choosing the source vector

$$\eta^a_\alpha(x) = \delta^{a a_i} \delta_{\alpha\alpha_i} \delta_{x, x_i}$$  (3.50)

with the index $i$ indicating some initial configuration of spinor, colour and position, the solution is seen to consist of the $(a_i, \alpha_i, x_i)$-th column of the inverse matrix,

$$V^c_{\gamma}(z) = (M^{-1})^{ca}_{\gamma\alpha i}(z, x_i).$$  (3.51)
Chapter 3. Simulation techniques

This solution describes the propagation from an initial point \(x_i\) to all other points \(z\) on the lattice. Using sources with different initial colour and spinor, one is able to generate the complete quark propagators appearing in Eq. 3.24 and two of the propagators which appear in the three-point function, Eq. 3.47. For these propagators we only need to obtain the columns of the inverse of the matrix labelled with \(x_i\).

The remaining propagator however, needs a second inversion because it does not originate from \(x_i\). Instead of calculating the propagator from every possible final position (which would require knowledge of the complete inverse of \(M\)), we choose a different source which leads to a combination of propagators and prevents us from inverting the complete matrix. This modified source reads

\[
\eta^a_{\alpha}(x) = \gamma^5_{\beta\epsilon} (M^{-1})^{ba}_\epsilon(x, x_i) \delta_{\epsilon, \epsilon} e^{iP \cdot x}
\]  

(3.52)

and is seen to give the solution

\[
V^\gamma(z) = \sum_{x_f} (M^{-1})^{cb}_{\beta\epsilon} (z, x_f) \gamma^5_{\beta\epsilon} (M^{-1})^{ba}_\epsilon(x_f, x_i) e^{-iP \cdot x_i}.
\]  

(3.53)

This is exactly the combination of propagators appearing in Eq. 3.47. It should be noted that this choice for the source implies that the momentum of the outgoing pion is fixed.

Thus to obtain the two-point function, we need to solve twelve (one for every spinor-colour combination) equations of the type in Eq. 3.48 with source 3.50. For the three-point function, we need to solve an additional twelve with source 3.52. It is important to note that this source corresponds to an unsmeared operator at the sink. It is quite straightforward to include the smearing in the source function, but the second inversion has to be repeated for every different value of \(R\), Eq. 3.26. Therefore the optimal smearing level has to be determined before one sets out to calculate the three-point function. In the next chapter we will see how one can use the two-point function for this purpose.

The inversion can be accelerated by preconditioning of the matrix. In our program we used even-odd preconditioning [57]. The idea is to divide the lattice in even and odd points (like a chessboard) and calculate first the inverse on the even points, where after the inverse on the odd points can be constructed from the even ones. The system of equations can thus be written as

\[
\begin{pmatrix}
A_{ee} & -\kappa \Delta_{eo} \\
-\kappa \Delta_{oe} & A_{oo}
\end{pmatrix}
\begin{pmatrix}
V_e \\
V_o
\end{pmatrix}
= \begin{pmatrix}
\eta_e \\
\eta_o
\end{pmatrix},
\]  

(3.54)

where the matrices \(A\) and \(\Delta\) are defined in Eq. 2.27 and colour and spinor indices are suppressed. Multiplying with

\[
\begin{pmatrix}
1_{ee} & \kappa \Delta_{eo} A_{oo}^{-1} \\
0 & 0
\end{pmatrix},
\]  

(3.55)

34
we find for the even sub-lattice
\[
(A_{ee} - \kappa^2 \triangle_{eo} A_{oo}^{-1} \triangle_{oe}) V_e = \eta_e + \kappa \triangle_{eo} A_{oo}^{-1} \eta_o.
\]
This defines a modified linear system of equations, but the column vector is still a column of the inverse of the old matrix (i.e. the even points thereof). Our task has thus been reduced to inverting a matrix which is half the size of the original one, although no saving of computing time has yet been achieved. The column vector of the odd points can be reconstructed from the even solution by
\[
V_o = A_{oo}^{-1} (\eta_o + \kappa \triangle_{oe} V_e).
\]
A small extra effort of this procedure is necessary for the inversion of $A_{oo}$. Nonetheless, the computer time needed for the inversion has been reduced significantly, since this preconditioning has produced a matrix with much smaller off-diagonal elements (due to the factor $\kappa^2$ in Eq. 3.56).

The matrix $A$ is diagonal in space. To invert this matrix, one can decompose it according to
\[
A = L^\dagger D L
\]
with $L$ a upper triangle matrix and $D$ a diagonal one in colour-Dirac indices. The inverse is then simply given by $L^{-1} D^{-1} L^\dagger$. For a more thorough discussion of the inversion of the fermion matrix, see [58,59]

### 3.7. Simulation parameters

In this section an overview of the simulation parameters will be given. In the case of zero temperature, a lattice of $24^3 \times 32$ will be used. We obtained the gluon configurations after an initial thermalisation of 2500 sweeps, after which we generated $O(100)$ configurations at intervals of 500 sweeps. One sweep consists of one heat bath step and 4 overrelaxation steps. We used periodic boundary conditions except for the time direction, where we used anti-periodic ones for the fermions. We have obtained results for 5 different $\kappa$-values, 0.13230, 0.13330, 0.13380, 0.13430, and 0.13480

corresponding to quark masses ranging from about 20 - 160 MeV. To solve the system of linear equations appearing in Sec. 3.6, we used the BiCGstabI routine [60]. The other parameters used are listed in table 3.1.

For finite temperature, the lattice size will be $32^3 \times 8$. At $\beta = 6.0$, this corresponds to a temperature of 0.93 $T_c$ [61]. Since fluctuations are expected to increase in the vicinity of the phase transition, we generated $O(200)$ configurations. Boundary conditions are kept the same. The other parameters can (and in certain ways must) be held fixed in order to be able to make a sensible comparison. The reason we changed the spatial lattice size in going from $T = 0$ to $T = 0.93 T_c$ will be given in Sec. 6.3.
### Table 3.1: Simulation parameters as used in this work.

<table>
<thead>
<tr>
<th>parameter</th>
<th>value</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta = 6/g_0^2$</td>
<td>6.0</td>
<td>coupling constant</td>
</tr>
<tr>
<td>$c_{SW}$</td>
<td>1.76923 [56]</td>
<td>Sheikholeslami-Wohlert improvement constant</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>0.5</td>
<td>link/staple mixing</td>
</tr>
<tr>
<td>$n$</td>
<td>4</td>
<td>number of fuzzing iterations</td>
</tr>
<tr>
<td>$R$</td>
<td>0 – 10</td>
<td>smearing level for the mesons</td>
</tr>
<tr>
<td>$Z_V^0$</td>
<td>0.77 [62]</td>
<td>renormalisation constant</td>
</tr>
<tr>
<td>$b_V$</td>
<td>1.52 [62]</td>
<td>renormalisation constant</td>
</tr>
<tr>
<td>$c_V$</td>
<td>$-0.107$ [62]</td>
<td>vector current improvement constant</td>
</tr>
</tbody>
</table>