

# 1

## Lattice Basics

---

In this section we want to present the basic concepts of lattice field theory, which has developed to a highly sophisticated research area in the last decades especially in the regime of QCD. This approach to field theory is not only of major concern for performing numerical calculations. It even provides a natural attempt to regularization, which makes it also an interesting topic for theorists in the field of gauge theories.

### 1.1 Discretizing the Lattice

A computer is not able to deal with continuous variables, since already a continuous real interval can contain an infinite amount of points, but a computer only has limited memory. Hence we need to limit the amount of points that we consider to make calculations. Since all the target space coordinates<sup>1</sup> depend on the worldsheet coordinates  $\tau$  and  $\sigma$ , it seems natural to discretize the worldsheet map. We therefore introduce a lattice spacing  $a$  and the lattice size in spacial direction  $L$  and in time direction  $T$ . Following the remarks in [1] we define the two-dimensional lattice  $\Lambda$  to be

$$\Lambda = \{(n_0, n_1) | n_0 = 0, 1, \dots, (T-1); n_1 = 0, 1, \dots, (L-1)\}. \quad (1.1.1)$$

With that we can express the worldsheet coordinates with the lattice coordinates

$$(\tau, \sigma) = (an_0, an_1). \quad (1.1.2)$$

And since the lattice spacing is overall constant we can abbreviatly refer to a field  $\Phi$  in the theory as a function of the lattice variables  $\Phi(n_0, n_1)$ . We now further want to discretize differential operators and integrals. The differentiation can be discretized in various ways inspired by its mathematical definition via differential quotients. To see additionally which order of error is introduced compared to the exact derivative, we examine the TAYLOR expansion of a real function  $f(x)$  with a small offset  $\epsilon$

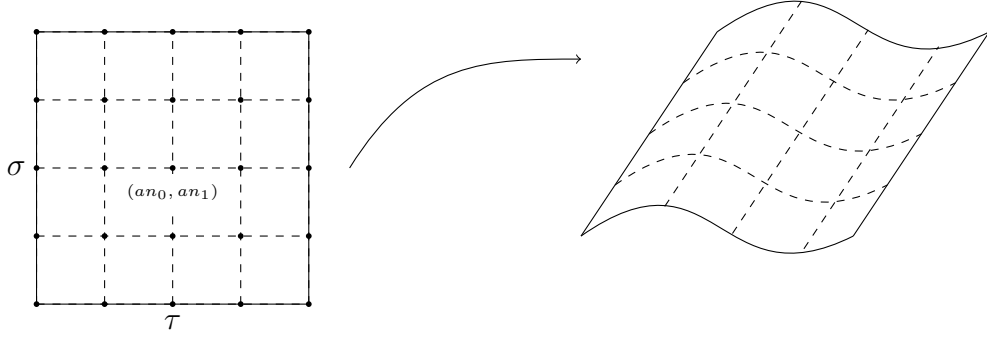
$$f(x \pm \epsilon) = f(x) \pm \epsilon f'(x) + \frac{\epsilon^2}{2} f''(x) \pm \frac{\epsilon^3}{6} f'''(x) + \dots \quad (1.1.3)$$

Now we can define a forward derivative as the finite difference

$$\frac{f(x + \epsilon) - f(x)}{\epsilon} = f'(x) + \mathcal{O}(\epsilon) \quad (1.1.4)$$

---

<sup>1</sup>which represent the fields in our two-dimensional QFT



**Figure 1.1.1** Mapping of the worldsheet grid into target space.

and in the same way also a backward derivative by using the  $f(x - \epsilon)$  term. Combining both leads to the symmetric derivative with an error of higher order in  $\epsilon$

$$\frac{f(x - \epsilon) - f(x + \epsilon)}{2\epsilon} = f'(x) + \mathcal{O}(\epsilon^2). \quad (1.1.5)$$

It is more convenient to use the symmetric finite difference, because the error introduced by the finite lattice spacing is hereby reduced by one order. By adding additional terms of  $f(x \pm 2\epsilon)$  it is possible to get a finite difference derivative which is exact up to an order of  $\mathcal{O}(\epsilon^3)$  and so on. Depending on the demanded accuracy one is able to regulate the introduced error via this method. With  $\epsilon$  being our lattice spacing  $a$  we find the derivative of our fields at  $n = (n_0, n_1)$  to be<sup>2</sup>

$$\partial_\alpha \Phi(\tau, \sigma) = \hat{\partial}_\alpha \Phi(n) + \mathcal{O}(a^2) \equiv \frac{\Phi(n - \hat{\alpha}) - \Phi(n + \hat{\alpha})}{2a} + \mathcal{O}(a^2), \quad (1.1.6)$$

$$\text{with} \quad \partial_\alpha \Phi(\tau, \sigma) \equiv \frac{\partial \Phi(\tau, \sigma)}{\partial \sigma_\alpha}, \quad \sigma_\alpha = (\tau, \sigma), \quad \alpha = 0, 1 \quad (1.1.7)$$

and where  $\hat{\alpha}$  is the unit vector in  $\alpha$ -direction. For the integration over the worldsheet coordinates we replace the integral by a finite sum over all lattice points

$$\int d\tau d\sigma \longrightarrow a^2 \sum_{n \in \Lambda}. \quad (1.1.8)$$

For a better visualization it is more convenient to use an abbreviated notation. In the following we will denote  $\phi \equiv \phi(n)$  as a field which depends on the lattice coordinates  $n$ . The same is true for an operator  $M \equiv M(m, n)$ .

<sup>2</sup>With  $\hat{\partial}_\alpha$  beeing the symmetric finite difference in  $\alpha$ -direction. We also define the forward and backward finite difference respectively by

$$\vec{\partial}_\alpha \Phi(n) = \frac{\Phi(n + \hat{\alpha}) - \Phi(n)}{a} \quad \text{and} \quad \bar{\partial}_\alpha \Phi(n) = \frac{\Phi(n) - \Phi(n - \hat{\alpha})}{a}.$$

A shorter expression of a matrix-vector like product is given by

$$\phi^T M \psi \equiv \sum_{m,n \in \Lambda} \phi(m) M(m,n) \psi(n). \quad (1.1.9)$$

To implement the previous relation on a computer as an actual matrix-vector product it is necessary to map the grid index  $n = (n_0, n_1)$  to a lexicographic vector index

$$l(n) = Ln_0 + n_1 \quad \text{with} \quad l = 0, 1, \dots, (V-1), \quad V = LT, \quad (1.1.10)$$

so that the according fields  $\phi_{l(n)} \equiv \phi(n)$  are actual vectors. For the finite differences derivatives we also need to know the neighboring relations. So if we want to move on the grid in direction  $\pm \hat{\alpha}$ , which lexicographic index corresponds to this movement? These relations are implemented into a *next neighbor* function (nb) which also takes into account the boundary conditions (toroidal in our case) and returns the corresponding vector index

$$l_{\pm \hat{\alpha}} = \text{nb}(l, \pm \hat{\alpha}). \quad (1.1.11)$$

We now want to define a symmetric finite difference without the factor  $1/a$  for the lexicographic notation by

$$\bar{\Delta}_\alpha \phi_l \equiv \frac{1}{2} (\phi_{l_{+\hat{\alpha}}} - \phi_{l_{-\hat{\alpha}}}). \quad (1.1.12)$$

In consequence we can write  $\bar{\Delta}_\alpha$  as a  $V \times V$  matrix

$$(\bar{\Delta}_\alpha)_{lp} = \frac{\delta_{l_{+\hat{\alpha}},p} - \delta_{l_{-\hat{\alpha}},p}}{2}. \quad (1.1.13)$$

For other than periodic boundary conditions some of the signs have to be flipped according to the relation (??). In the same way we can define forward and backward finite differences by

$$(\Delta_\alpha)_{lp} = \delta_{l_{+\hat{\alpha}},p} - \delta_{l,p}, \quad (\Delta_\alpha^*)_{lp} = \delta_{l,p} - \delta_{l_{-\hat{\alpha}},p}. \quad (1.1.14)$$

We will come back to this notation in ???. For now on we will presume using the grid notation for the further studies.

## 1.2 Bosons and Fermions on the Lattice

### 1.2.1 Bosonic Propagator

In the examined theory bosons and fermions are both scalar fields of either real, or complex GRASSMANN type. The equations of motion of bosons are given by the KLEIN-GORDON equation, which contains only second derivatives. To discretize the second derivative we define the corresponding

finite difference operator simply by applying a forward and a backward finite difference

$$\hat{\partial}_\alpha^2 \Phi(n) = \tilde{\partial}_\alpha \vec{\partial}_\alpha \Phi(n) = \frac{\Phi(n + \hat{\alpha}) - 2\Phi(n) + \Phi(n - \hat{\alpha})}{a^2}. \quad (1.2.1)$$

In a Lagrangian we would find the differential operator enclosed by the fields, for instance like

$$\Phi(n) (\hat{\partial}_\alpha^2 \Phi)(n) = \sum_{m \in \Lambda} \Phi(n) \hat{\partial}_\alpha^2(n, m) \Phi(m), \quad (1.2.2)$$

where we use (1.2.1) to define

$$\hat{\partial}_\alpha^2(n, m) \equiv \frac{\delta_{m, n + \hat{\alpha}} - 2\delta_{m, n} + \delta_{m, n - \hat{\alpha}}}{a^2}. \quad (1.2.3)$$

To investigate the behaviour of bosons under the discretization we are interested in the propagator, which is the inverse of the differential operator in momentum space. We therefore perform a discrete FOURIER transform, which is discussed in Appendix ???. Both indices  $m$  and  $n$  transform independently and in order to perform a unitary similarity transformation the second index is FOURIER transformed using the complex conjugated phase [1]

$$\begin{aligned} \tilde{\partial}_\alpha^2(p, q) &= \frac{1}{|\Lambda|} \sum_{m, n \in \Lambda} e^{-iq \cdot na} \hat{\partial}_\alpha^2(n, m) e^{ip \cdot ma} \\ &= \frac{1}{|\Lambda|} \sum_{n \in \Lambda} e^{-i(q-p) \cdot na} \frac{1}{a^2} [e^{ip_\alpha a} - 2 + e^{-ip_\alpha a}] \\ &= \delta(p - q) \tilde{\partial}_\alpha^2(p), \end{aligned} \quad (1.2.4)$$

where the FOURIER transform of the second derivative is defined by

$$\tilde{\partial}_\alpha^2(p) \equiv -\frac{4}{a^2} \sin^2 \left( \frac{p_\alpha a}{2} \right). \quad (1.2.5)$$

The EUCLIDEAN KLEIN-GORDON operator  $\tilde{D}(p)$  in momentum space with particle mass  $m$  reads

$$\tilde{D}(p) = m^2 + \sum_{\alpha=0}^1 \tilde{\partial}_\alpha^2(p). \quad (1.2.6)$$

And so we find the propagator simply by inverting the previous relation. Of special interest is the massless case, where we want to investigate the continuum limit

$$\tilde{D}^{-1}(p) \Big|_{m=0} = \frac{1}{-\frac{4}{a^2} \sum_{\alpha=0}^1 \sin^2 \left( \frac{p_\alpha a}{2} \right)} \xrightarrow{a \rightarrow 0} -\frac{1}{p^2}. \quad (1.2.7)$$

This is precisely what we expected for the EUCLIDEAN propagator in the continuum. Since we have  $p_\alpha \in (-\frac{\pi}{a}, \frac{\pi}{a}]$ , there are no additional poles within the sine function other than the physical one at  $p = (0, 0)$ .

### 1.2.2 Fermionic Propagator and the Doubling Problem

In the case of fermions the equations of motion in 4-dimensional QFT are given by the Dirac equation, which is a linearisation of the KLEIN-GORDON equation and only contains first derivatives. To not explicitly break existing symmetries, we have to go with the symmetric derivative in this case. Now we can follow the exact same steps as for the bosonic case to investigate the continuum limit of a DIRAC like operator in our 2-dimensional space time. As mentioned we start from the symmetric finite difference

$$\widehat{\partial}_\alpha(m, n) \equiv \frac{\delta_{m, n+\hat{\alpha}} - \delta_{m, n-\hat{\alpha}}}{2a} \quad (1.2.8)$$

and proceed with the FOURIER transform

$$\widetilde{\partial}_\alpha(p, q) = \frac{1}{|\Lambda|} \sum_{m, n \in \Lambda} e^{-iq \cdot na} \widehat{\partial}_\alpha(n, m) e^{ip \cdot ma} \quad (1.2.9)$$

$$= \frac{1}{|\Lambda|} \sum_{n \in \Lambda} e^{-i(q-p) \cdot na} \left[ \frac{e^{ip_\alpha a} - e^{-ip_\alpha a}}{2a} \right] \quad (1.2.10)$$

$$= \delta(p - q) \widetilde{\partial}_\alpha(p), \quad (1.2.11)$$

here we are left with the following differential operator in momentum space

$$\widetilde{\partial}_\alpha(p) \equiv \frac{i}{a} \sin(p_\alpha a). \quad (1.2.12)$$

To construct an EUCLIDEAN DIRAC like operator in two dimensions we use the first two PAULI matrices<sup>3</sup>

$$\widetilde{D}_D(p) = m\mathbb{1} + \sum_{\alpha=0}^1 \gamma_\alpha \widetilde{\partial}_\alpha(p) = m\mathbb{1} + \frac{i}{a} \sum_{\alpha=0}^1 \gamma_\alpha \sin(p_\alpha a). \quad (1.2.13)$$

The inverse propagator follows from the properties of the PAULI matrices to be

$$\widetilde{D}_D^{-1}(p) = \frac{m\mathbb{1} - \frac{i}{a} \sum_{\alpha=0}^1 \gamma_\alpha \sin(p_\alpha a)}{m^2 + a^{-2} \sum_{\alpha=0}^1 \sin^2(p_\alpha a)}. \quad (1.2.14)$$

Once more we want to study the continuum limit of the propagator for the massless case and find by setting  $m = 0$

$$\widetilde{D}_D^{-1}(p) \Big|_{m=0} = \frac{-\frac{i}{a} \sum_{\alpha=0}^1 \gamma_\alpha \sin(p_\alpha a)}{a^{-2} \sum_{\alpha=0}^1 \sin^2(p_\alpha a)} \xrightarrow{a \rightarrow 0} \frac{-i \sum_{\alpha=0}^1 \gamma_\alpha p_\alpha}{p^2}. \quad (1.2.15)$$

Here again the propagator has the correct naive continuum limit but this time we face another problem. Since again  $p_\alpha$  is in the domain  $(-\frac{\pi}{a}, \frac{\pi}{a}]$ ,

<sup>3</sup>

$$\gamma_0 = \sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \gamma_1 = \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

there occur three other non-physical poles within the sine function at  $p = (0, \pi/a)$ ,  $p = (\pi/a, 0)$  and  $p = (\pi/a, \pi/a)$ . As a result in the numerical simulation it seems that there are in this case three additional fermions, which emerge only from the method of discretization and are non-physical. This is obviously leading to distorted calculations and needs to be circumvented. That particular problem is known as fermion doubling and has been investigated extensively by WILSON, who introduced a possible solution. He suggested to add a term proportional to the discretized LAPLACE operator[2], which is known as a WILSON term. We can express it in momentum space with the help of (1.2.5) and find<sup>4</sup>

$$W(p) \equiv -\mathbb{1} \frac{a}{2} \sum_{\alpha=0}^1 \tilde{\partial}_{\alpha}^2(p) = \mathbb{1} \frac{1}{a} \sum_{\alpha=0}^1 (1 - \cos(p_{\alpha}a)). \quad (1.2.16)$$

Since the LAPLACE operator is of order  $\mathcal{O}(a^0)$ , we see that the WILSON term vanishes in the continuum limit. It also vanishes for  $p_{\alpha} = 0$  and therefore gives no contribution to the physical pole of the theory. If we now invert the new DIRAC-WILSON operator

$$\tilde{D}_W(p) \equiv \tilde{D}_D(p) + W(p), \quad (1.2.17)$$

we find the new massless momentum space propagator

$$\tilde{D}_W^{-1} \Big|_{m=0} = \frac{\mathbb{1} \frac{1}{a} \sum_{\alpha=0}^1 (1 - \cos(p_{\alpha}a)) - \frac{i}{a} \sum_{\alpha=0}^1 \gamma_{\alpha} \sin(p_{\alpha}a)}{\left(\frac{1}{a} \sum_{\alpha=0}^1 (1 - \cos(p_{\alpha}a))\right)^2 + \frac{1}{a^2} \sum_{\alpha=0}^1 \sin^2(p_{\alpha}a)}. \quad (1.2.18)$$

Here we can convince ourselves, that there are no more additional poles in the domain of  $p_{\alpha}$ , since for the points where  $p_{\alpha} = \pi/a$  the denominator does not vanish like before. Actually for these points the WILSON-term behaves like an additional mass term that is growing infinitely in the continuum limit and consequently decouples from the theory.

## 1.3 Monte Carlo Methods

Now after we know, how to discretize differential operators and act with them on bosonic and fermionic quantities, we are able to build a discretized Lagrangian as well. But what we really need is a method to perform the calculation of observables in the path integral formalism. In QFT in the continuum we calculate the expectation value of an observable  $O$  via [3]

$$\langle O \rangle = \frac{1}{Z} \int \mathcal{D}\phi O(\phi) \exp(-S[\phi]), \quad (1.3.1)$$

---

<sup>4</sup>using the identity

$$\begin{aligned} \cos(2x) &= \cos^2(x) - \sin^2(x) \\ \Leftrightarrow \sin^2(x) &= \frac{1}{2} (1 - \cos(2x)) \end{aligned}$$

where  $S$  is the EUCLIDEAN action depending on the collection of fields  $\phi$  and  $Z$  is the partition function

$$Z = \int \mathcal{D}\phi \exp(-S[\phi]). \quad (1.3.2)$$

In fact there is no rigorous mathematical definition for the path integral in the continuum. It is motivated from a discretized measure, which is then somehow said to be made continuous. This step is considered ill defined by most mathematicians, but since we stay on a discretized lattice, the path integral measure  $\mathcal{D}\phi$  can be perfectly well defined and is represented by

$$\mathcal{D}\phi = \prod_{n \in \Lambda} d\phi(n). \quad (1.3.3)$$

This means, that for every lattice point a multidimensional integral needs to be calculated. Even for small lattices this task becomes tortuous and soon even supercomputers would not be able to do that calculation in reasonable time. Out of this reason other methods of calculation have been considered. The most favoured one is a statistical method going under the name of Monte Carlo simulation. With this type of calculation it is possible to obtain good statistical approximations of high-dimensional integrals in reasonable time. At this point we want to study some details of the Monte Carlo methods used in the scope of this thesis.

### 1.3.1 Monte Carlo Integration

The easiest way to understand the idea behind any form of calculation with Monte Carlo methods is via the Monte Carlo integration. Suppose we want to calculate the following multi-dimensional integral

$$I = \int_{\Omega} f(x) dx, \quad (1.3.4)$$

where  $\Omega \subset \mathbb{R}^n$  is a compact integration domain. Then probability theory tells us [4], that we can approximate the integral  $I$  with a set of random numbers  $\{\bar{x}_1, \dots, \bar{x}_N\}$  uniformly distributed on  $\Omega$  with

$$I_N = V \frac{1}{N} \sum_{i=1}^N f(\bar{x}_i). \quad (1.3.5)$$

Hereby  $V = \int_{\Omega} dx$  denotes the integration volume. One can show that  $I_N$  approximates  $I$  up to an error of order  $\mathcal{O}(1/\sqrt{N})$  and the law of large numbers tells us, that  $\lim_{N \rightarrow \infty} I_N = I$ . One can also write for the expectation value  $\langle f \rangle$  of  $f$ , that

$$\langle f \rangle = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N f(\bar{x}_i) = \frac{\int_{\Omega} f(x) dx}{\int_{\Omega} dx}. \quad (1.3.6)$$

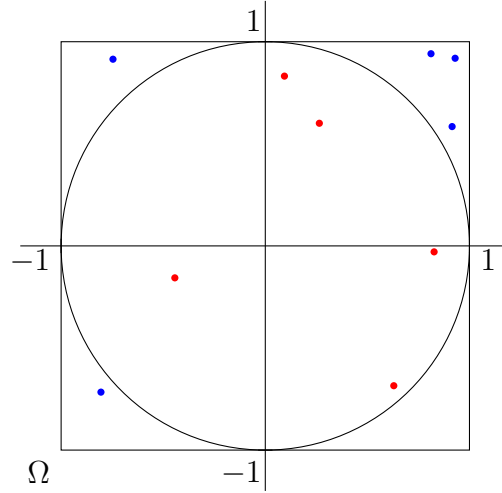
An instructive example is to calculate the value of  $\pi$  with the 2D-integral  $I = \int_{\Omega} H(x, y) dx dy$  over the domain  $\Omega = [-1, 1] \times [-1, 1]$  with the function

$$H(x, y) = \begin{cases} 1 & \text{if } x^2 + y^2 \leq 1 \\ 0 & \text{else.} \end{cases} \quad (1.3.7)$$

Analytically we know that  $I$  corresponds to the area of the unit circle and has the value  $I = \pi$ . With the volume being  $V = 4$  we can find an estimate

$$\pi \approx \frac{4}{N} \sum_{i=1}^N H(\bar{x}_i). \quad (1.3.8)$$

This means that we can estimate the value of  $\pi/4$  by counting the number of points lying within the circle (see figure 1.3.1) and dividing by the total number of points.



**Figure 1.3.1** Graphic of random numbers uniform in  $\Omega$ .  $\pi/4$  can be estimated by the number of red points within the circle divided by total number of points.

### 1.3.2 Importance Sampling and Markov Chains

More generally one can estimate integrals over a function  $f(x)$  multiplied with a certain weight function  $\rho(x)$  in the same way. We can write for the expectation value

$$\langle f \rangle_{\rho} \equiv \frac{\int_{\Omega} f(x) \rho(x) dx}{\int_{\Omega} \rho(x) dx} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N f(\bar{x}_i). \quad (1.3.9)$$

But in this case the random numbers  $\bar{x}_i$  need to be distributed through the weight function  $\rho(x)$ . For this to be possible  $\rho(x)$  needs to be a valid probability density function. Comparing with (1.3.1) we can convince ourselves, that this is the case for the EUCLIDEAN path integral. We can exploit



this fact to make our calculation more efficient. In principle one could go with the simple Monte Carlo integration scheme, but since our integration domain is no compact interval but the  $\mathbb{R}^n$ , we would produce lots of configurations that would give no contribution to the path integral due to the weight factor and therefore introduce a large error. So if we produce random configurations distributed through a probability density given by the weight factor, we would generate mostly configurations with a non vanishing contribution and the statistical error will be reduced. This procedure is called importance sampling. But to do the importance sampling one needs to be able to produce random configurations according to the given probability density. Since the action appearing in this probability density is in general a very complicated function, this task becomes highly non trivial and there usually is no way to directly transform uniformly distributed random configurations into the favoured form. To make this more explicit, consider the one dimensional integral with normalized probability density function [5]

$$\langle f \rangle_\rho = \int_a^b f(x) \rho(x) dx, \quad \int_a^b \rho(x) dx = 1. \quad (1.3.10)$$

The distribution function  $P(x)$  is given by

$$P(x) = \int_a^x \rho(t) dt. \quad (1.3.11)$$

If we perform the substitution

$$y(x) \equiv P(x), \quad \text{with} \quad \rho(x) = P'(x) = y'(x), \quad y \in [0, 1], \quad (1.3.12)$$

we are left with

$$\langle f \rangle_\rho = \int_a^b f(x) y'(x) dx = \int_0^1 f(x(y)) dy, \quad (1.3.13)$$

where  $x(y) = P^{-1}(y)$ . So if we generate uniformly distributed random numbers  $\bar{y}$  and use  $\bar{x} = P^{-1}(\bar{y})$ , which is now distributed according to  $\rho(x)$ , we can write

$$\langle f \rangle_\rho = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N f(\bar{x}_i). \quad (1.3.14)$$

For this to be true  $P(x)$  needs to be strongly increasing and globally invertible. Extended to the higher-dimensional problem of path integrals this demand is met very unlikely, as said before.

To circumvent this issue one uses a statistical model called MARKOV chain to obtain configurations that are distributed according to the BOLTZMANN factor of the path integral. The idea is to start from an arbitrary configuration and generate a new one from the previous configuration via

a pseudo random number generation process. Thereby new configurations are generated in a stochastic sequence that eventually follows an equilibrium distribution  $P(\phi)$  [1]. The configurations in the chain can be numbered by an index, the so called Monte Carlo time

$$\phi_0 \rightarrow \phi_1 \rightarrow \dots \rightarrow \phi_{N_{\text{th}}} \rightarrow \dots \quad (1.3.15)$$

The first configurations up to an index  $N_{\text{th}}$  should not be used for measurements since they are most certainly not properly distributed. The process needs a specific amount of time to reach its equilibrium distribution. Therefore this period is called equilibration or thermalization phase. A MARKOV process is characterized by a conditional transition probability  $T(\phi'|\phi)$  to move from a configuration  $\phi$  to  $\phi'$ . This probability depends only on the configurations  $\phi$  and  $\phi'$  and not any previous ones. It obeys the properties

$$0 \leq T(\phi'|\phi) \leq 1, \quad \sum_{\phi} T(\phi'|\phi) = 1. \quad (1.3.16)$$

An important restriction is that the process must not get stuck at any configuration. It must always be possible to move between two specified configurations within a finite Monte Carlo time. This can be realised if the transition probability is strictly positive between all states which is called strong ergodicity. Furthermore the probability to move to  $\phi'$  from any other state must be the same as to leave  $\phi'$  to any other state which can be condensed to the following balance equation

$$\sum_{\phi} T(\phi'|\phi)P(\phi) = \sum_{\phi} T(\phi|\phi')P(\phi'). \quad (1.3.17)$$

A special solution to this equation will be obtained if one assumes that the equation holds for every term of the sum

$$T(\phi'|\phi)P(\phi) = T(\phi|\phi')P(\phi'). \quad (1.3.18)$$

This condition is called detailed balance and is used by most algorithms generating MARKOV chains.

### 1.3.3 Fermions on the Lattice and the Hybrid Monte Carlo Algorithm

There are many possible algorithms to generate field configurations. The choice is depending on the structure of the problem at hand. For the here presented case the algorithm of choice is called rational hybrid Monte Carlo (RHCM). The basic reason to use this algorithm is the presence of fermions in a specific manner. It is in general complicated to deal with fermions in numerical simulations due to the anti commuting properties of GRASSMANN variables. An efficient procedure is to integrate out fermions via GRASSMANN integration, presented in Appendix ??, which results in our case in

the Pfaffian of a fermionic operator  $M[\phi]$  depending on the bosonic fields  $\phi$ . We now want to include the Pfaffian as a probability weight factor in the process of generating MARKOV chain configurations. To act as a probability weight factor,  $\text{Pf } M$  must be real and nonnegative. If we consider this to be the case for the moment, we can write

$$\text{Pf } M = (\det M)^{\frac{1}{2}} = (\det MM^\dagger)^{\frac{1}{4}}. \quad (1.3.19)$$

In general one could have to deal with an arbitrary fraction  $\alpha$  in the exponent on the right side of (1.3.19). If we restrict  $\text{Pf } M$  to be also nonzero and therefore entirely positive using that  $MM^\dagger$  is positive definite, one can rewrite the determinant as a bosonic path integral with pseudofermions  $\eta$

$$(\det MM^\dagger)^\alpha = \frac{1}{\det (MM^\dagger)^{-\alpha}} \sim \int \mathcal{D}\eta \mathcal{D}\eta^\dagger e^{-\eta^\dagger (MM^\dagger)^{-\alpha} \eta}. \quad (1.3.20)$$

Since there is an inversion included, the operator in the exponent is highly nonlocal in the bosonic fields  $\phi$ . So even a small change in the fields could cause a large difference in the weight factor. Ergo one needs an algorithm that ensures that the generated configurations are part of the contributing regime of the weight factor. In a case like that it proved efficient to use a hybrid Monte Carlo (HMC) algorithm when  $\alpha = 1$ , which uses HAMILTONIAN dynamics trajectories along some fictitious time to generate new configurations. For non-integer  $\alpha$  one can use a rational approximation of  $(MM^\dagger)^{-\alpha}$ . Combined with the HMC this is called the rational hybrid Monte Carlo (RHMC) algorithm.

At this point we want to introduce the HMC algorithm which has been proposed in [6]. A more detailed explanation is presented in [1, 2] which serves as a guideline for the following. The HMC is referred to as hybrid because it combines a molecular dynamic (MD) evolution with random noise patterns to simulate quantum fluctuations, an additional METROPOLIS acceptance step lets it result into an exact MC algorithm. The field update is separated into two phases. Pseudofermions are updated first with an exact heat bath, while the bosons are kept constant. Therefore, a configuration  $\xi$  is generated from a GAUSSIAN distribution  $P(\xi) \sim e^{-\xi^\dagger \xi}$  and we set  $\eta = M\xi$ . In the second phase a molecular dynamic evolution is used to update the bosonic fields while the pseudofermions are kept constant. We can now think of the effective action  $S_{\text{eff}}[\phi] = S_B[\phi] + \eta^\dagger (MM^\dagger)^{-1} \eta$  as only dependent on the bosonic fields  $\phi$ . Now for each field  $\phi_i$  a conjugate momentum  $\pi_i$  is introduced and we expand the path integral with  $\int \mathcal{D}\pi \exp(-1/2 \pi^2)$  to obtain

$$\langle O \rangle = \frac{\int \mathcal{D}\phi \mathcal{D}\eta \mathcal{D}\pi O(\phi) e^{-H[\phi, \pi]}}{\int \mathcal{D}\phi \mathcal{D}\eta \mathcal{D}\pi e^{-H[\phi, \pi]}}, \quad (1.3.21)$$

where we have introduced the HAMILTONIAN

$$H[\phi, \pi] = \frac{1}{2} \pi^2 + S_{\text{eff}}[\phi], \quad \pi^2 = \sum_i \sum_{n \in \Lambda} \pi_i^2(n). \quad (1.3.22)$$

The evolution of the fields follows according to the classical equations of motion along a fictitious MC time  $\tau$

$$\begin{aligned}\dot{\pi}_i &= -\frac{\partial H}{\partial \phi_i} = -\frac{\partial S_{\text{eff}}}{\partial \phi_i} \\ \dot{\phi}_i &= \frac{\partial H}{\partial \pi_i} = \pi_i\end{aligned}\tag{1.3.23}$$

A numerical implementation of (1.3.23) introduces a discrete step size  $\epsilon \equiv \delta\tau$  and errors of order  $\mathcal{O}(\epsilon^2)$ . An additional METROPOLIS acceptance step can correct these errors. First the momenta  $\pi$  are randomly generated from a GAUSSIAN distribution  $P_G(\pi) = \exp(-1/2 \pi^2)$ . Then a numerical integration scheme is used to change the configuration  $\{\phi, \pi\}$  along a discretized trajectory in phase space to the new point  $\{\phi', \pi'\}$ . The transition probability  $T_{\text{MD}}(\phi', \pi' | \phi, \pi)$  of the MD evolution depends on the integration scheme. The latter needs to be reversible and area preserving for the algorithm to obey detailed balance. Thus, a *leapfrog integration* scheme is used. For one trajectory the fields  $\phi$  are evolved in  $n$  steps of length  $\epsilon$ . The momenta start with half a step of length  $\epsilon/2$ , then  $(n-1)$  full steps are performed and again a half-step. The first half-step is given by

$$\begin{aligned}\pi_i(\tfrac{\epsilon}{2}) &= \pi_i(0) - \frac{\partial S_{\text{eff}}(\phi(0))}{\partial \phi_i} \frac{\epsilon}{2}, \\ \phi_i(\epsilon) &= \phi_i(0) + \pi_i(\tfrac{\epsilon}{2}) \epsilon.\end{aligned}\tag{1.3.24}$$

The next steps for  $j = 1, \dots, n-1$  are

$$\begin{aligned}\pi_i((j + \tfrac{1}{2}) \epsilon) &= \pi_i((j - \tfrac{1}{2}) \epsilon) - \frac{\partial S_{\text{eff}}(\phi(j\epsilon))}{\partial \phi_i} \epsilon \\ \phi_i((j + 1) \epsilon) &= \phi_i(j\epsilon) + \pi_i((j + \tfrac{1}{2}) \epsilon) \epsilon.\end{aligned}\tag{1.3.25}$$

With the last half-step we arrive at the final momentum

$$\pi_i(n\epsilon) = \pi_i((n + \tfrac{1}{2}) \epsilon) - \frac{\partial S_{\text{eff}}(\phi(n\epsilon))}{\partial \phi_i} \frac{\epsilon}{2}.\tag{1.3.26}$$

With an additional METROPOLIS acceptance step

$$T_A(\phi', \pi' | \phi, \pi) = \min(1, \exp(H[\phi, \pi] - H[\phi', \pi']))\tag{1.3.27}$$

the total transition probability to move from  $\phi$  to  $\phi'$  is given by

$$T(\phi' | \phi) = \int \mathcal{D}\pi' \mathcal{D}\pi T_A(\phi', \pi' | \phi, \pi) T_{\text{MD}}(\phi', \pi' | \phi, \pi) P_G(\pi).\tag{1.3.28}$$

This has been proved to obey detailed balance in [6]. Concluding one can summarize the HMC algorithm in the following steps:

- **Pseudofermions**

Generate the pseudofermion field  $\eta = M\xi$ , where  $\xi$  is distributed according to  $\exp(-\xi^\dagger \xi)$ .

- **Conjugate fields**

For an initial boson configuration  $\phi^{(0)}$  generate  $\pi^{(0)}$  according to the GAUSSIAN distribution  $\exp(-\frac{1}{2}\pi^2)$ .

- **Initial step**

$$\pi_i^{(\frac{1}{2})} = \pi_i^{(0)} - \frac{\epsilon}{2} F_i[\phi] \Big|_{\phi^{(0)}}.$$

- **Intermediate steps**

Full steps for  $j = 1, \dots, n-1$

$$\phi_i^{(j)} = \phi_i^{(j-1)} + \pi_i^{(j-\frac{1}{2})}, \quad \pi_i^{(j+\frac{1}{2})} = \pi_i^{(j-\frac{1}{2})} - \epsilon F_i[\phi] \Big|_{\phi^{(j)}}.$$

- **Final step**

$$\phi'_i = \phi_i^{(n)} = \phi_i^{(n-1)} + \pi_i^{(n-\frac{1}{2})}, \quad \pi'_i = \pi_i^{(n)} = \pi_i^{(n-\frac{1}{2})} - \frac{\epsilon}{2} F_i[\phi] \Big|_{\phi^{(n)}}.$$

- **Monte Carlo step**

Accept new configuration  $\phi'$  if a random number  $r \in [0, 1)$  is smaller than

$$\exp \left( \frac{1}{2} (\pi^2 - \pi'^2) + S_B[\phi] - S_B[\phi'] + \xi^\dagger \xi - \eta^\dagger (M' M'^\dagger)^{-1} \eta \right).$$

Here we used  $\phi_i^{(j)} \equiv \phi_i(j\epsilon)$  and the driving forces are denoted by

$$F_i[\phi] = \frac{\partial}{\partial \phi_i} \left[ S_B[\phi] + \eta^\dagger (M M^\dagger)^{-1} \eta \right]. \quad (1.3.29)$$

### 1.3.4 Numerical sign problems

As discussed previously we want to calculate expectation values of observables  $O(\phi)$ , depending on bosonic fields  $\phi$  which are distributed according to a certain probability weight function  $\rho(\phi)$

$$\langle O \rangle_\rho = \frac{\int \mathcal{D}\phi O(\phi) \rho(\phi)}{\int \mathcal{D}\phi \rho(\phi)}. \quad (1.3.30)$$

The feasibility of calculations like that via Monte Carlo methods is only guaranteed if the weight function  $\rho(\phi)$  is non-negative. In the process of dealing with fermions it is sometimes necessary to introduce bosonic auxiliary fields by a HUBBARD-STRATONOVICH (HS) transformation<sup>5</sup> which might lead to a complex phase in the weight function. For fluctuations of this phase that are very close to the purely real spectrum, this can be corrected by a reweighting procedure. If nonetheless the phase is distributed on the whole complex spectrum where the reweighting tends to fail, then one is talking of a *sign problem*. There are several approaches to fix this issue but none of them works as a general method to cure the sign problem.

<sup>5</sup>See Appendix ?? for further details on the HS transformation.



## Bibliography

---

- [1] C. Gattringer and C. Lang, “*Quantum Chromodynamics on the Lattice: An Introductory Presentation*”, Springer Berlin Heidelberg (2009).
- [2] I. Montvay and G. Münster, “*Quantum fields on a lattice*”, Cambridge University Press (1994).
- [3] M. E. Peskin and D. V. Schroeder, “*An introduction to quantum field theory*”, Addison-Wesley Pub. Co (1995).
- [4] M. H. Kalos and P. A. Whitlock, “*Monte Carlo Methods*”, 2 edition, Wiley-VCH (2008).
- [5] H. J. Rothe, “*Lattice Gauge Theories: An Introduction*”, 3 edition, World Scientific Publishing Company (2005).
- [6] S. Duane, A. Kennedy, B. J. Pendleton and D. Roweth, “*Hybrid Monte Carlo*”, Physics Letters B 195, 216 (1987).