Chapter 1

Introduction

In this project we will study scalar quantum field theory on discrete spaces with random fluctuations. After some general considerations concerning discrete spaces we will eventually settle on percolation lattices. The aim is to see whether such discrete spaces have a well-defined continuum limit and how the random structure of the underlying space affects the continuum theory.

We start by formulating a minimalist method to construct discrete spaces and write down actions for them. We then determine which of their properties we want to study and discuss some examples. Here we will introduce percolation lattices and another important example, the Bethe lattice. The most important quantity that we will study is the propagator.

In the next chapter we gather the necessary analytical and numerical tools to calculate the propagator and other quantities. We will discuss several methods, each with their own advantages and disadvantages.

We then start testing the available methods on lattices of different dimensions to see under which circumstances they yield reliable results and to check if the results are in agreement with each other.

Finally, after a lot of preparation, we are ready to apply the methods to percolation configurations. We discuss the results and compare them to the results on regular lattices.

Although this project originated from curiosity rather than utilitarianism, its results are relevant in the context of modern physics. Extrapolating results from quantum mechanics and general relativity it is expected that space(time) is irregular at very small length scales. That is, if we were able to ‘zoom in’ we would see that the space that we experience as smooth is actually fluctuating wildly at the microscopic level. In particle physics, however, this is usually neglected and theories are formulated against a smooth and unchanging spacetime background. In contrast, we will study a simple quantum field theory defined on a randomly fluctuating space and attempt to derive its behaviour in the continuum limit.

The main question is whether the continuum theory on the random lattice is
equivalent to one on a normal lattice or whether it behaves differently in a fundamental way.
Chapter 2

Theory

In this chapter we lay the theoretical foundations that will be used throughout the rest of this thesis. We first discuss the discretization of quantum field theories and subsequently develop a framework allowing us to write down an action for a general discrete space. We introduce percolation lattices and Bethe lattices and address the problem of defining distance on a general configuration.

2.1 Discrete Spaces

Quantum field theory is usually defined on a continuous space. There are two major reasons to move from such a continuum theory to one defined on a discrete space. Firstly, it enables one to do numerical calculations; a computer simply can’t handle an infinite amount of numbers. Secondly, a discrete formulation can provide a way to define the theory rigorously when this is troublesome on a continuous space. For example, in the path integral formulation of quantum field theory we have to consider contributions from all paths from one point to another. There is an uncountably infinite number of them and one can run into trouble writing this down in a mathematically precise manner. In a theory defined on a discrete space this is not a problem.

We can define a theory on a discrete space in such a way that it is a very good approximation of one on a continuous space. One can compare this to a television or computer screen: the two-dimensional surface of the screen is discrete (it is made up out of pixels) but the image on it appears smooth from a distance as long as the pixels are small enough.

An example is Euclidean scalar field theory in \( D \) dimensions, of which the continuum action is

\[
S_C = \int d^Dx \left[ \frac{1}{2} m^2 \phi^2 + \frac{1}{2} \left( \nabla \phi \right)^2 \right].
\]

We can discretize the above action by overlaying the \( D \)-dimensional space with a \( D \)-cubic lattice with lattice parameter \( \Delta \). We only consider the value of the
CHAPTER 2. THEORY

field $\phi(\vec{x})$ at lattice vertices, for which $\vec{x} = \Delta \vec{n}$ where the components of $\vec{n}$ are integers. We approximate the derivative in the $\hat{x}_i$-direction as

$$\partial_i \phi(\vec{x}) \approx \frac{\phi(\vec{x} + \Delta \hat{x}_i) - \phi(\vec{x})}{\Delta}.$$  

The integral is approximated by a sum:

$$\int d^D \vec{x} \to \Delta^D \sum_{n}.$$  

Combining the above operations, we find the discrete action

$$S = \sum_n \frac{1}{2} \lambda_2 \phi_n^2 + \sum_{\text{nbrs } m,n} \frac{1}{2} \xi (\phi_m - \phi_n)^2. \quad (2.1)$$

Note that we have introduced some new notation. First of all, the second sum is over all neighbouring pairs $(m, n)$. Furthermore, we have combined the vector label $\vec{n}$ into a new generic label $n$. We have also introduced the new parameters

$$\lambda_2 \equiv \Delta^D m^2$$

$$\xi \equiv \Delta^{D-2}.$$  

To go from the discrete to the continuum theory we should take the length scale $\Delta$ to be infinitesimally small:

$$S_{C} = \lim_{\Delta \to 0} S.$$  

We shall often use the fact that $\Delta$ is proportional to $\sqrt{\lambda_2/\xi}$ so that taking the continuum limit corresponds to taking $\lambda_2/\xi \to 0$.

2.2 Connecting the Dots

From the above example we see that the basic building blocks of a discrete quantum field theory are local interactions, pertaining to single points and neighbour interactions pertaining to pairs of points that we call neighbours. Each point only communicates directly with its neighbours.

In this view, a discrete space is nothing more than a collection of points with connections between them, where a connection signifies that two points are neighbours. Remember that in the above example we were able to arbitrarily label the points. We can do this because the label is irrelevant; the connections to other points determine a point’s position in space.

In mathematics such a structure of interconnected objects is called a graph\footnote{The word ‘graph’ is also (and more commonly) used for a ‘graphical representation of a function’ but that is not what we mean here!}. The points are also called nodes or vertices and the connecting lines are often called edges. To avoid ambiguity by the use of the word graph, and to emphasize
2.3. THE ACTION

The fact that we use graphs to represent spaces I will refer to a set of points with connections as a \textit{configuration}.

The above ideas lead us to the general observation that \textit{any configuration is equivalent to a discrete space on which a well-defined quantum field theory can be formulated}. Simply said, this means that we can draw a number of dots on a piece of paper, connect them with lines in an arbitrary way, and then write down an action. Each dot leads to \textit{local} terms in the action and each line leads to \textit{neighbour interaction} terms in the action.

As an example, consider the configuration depicted in (b). According to equation (2.1) the action corresponding to this configuration is

\[
S = \frac{1}{2} \lambda_2 \left[ \phi_0^2 + \phi_1^2 + \phi_2^2 + \phi_3^2 + \phi_4^2 \right] + \frac{1}{2} \xi \left[ (\phi_0 - \phi_1)^2 + (\phi_1 - \phi_2)^2 + (\phi_2 - \phi_3)^2 + (\phi_2 - \phi_4)^2 \right],
\]

where we have assigned arbitrary labels to the different nodes.

We need our configurations to be \textit{simple graphs}: two points can only be connected once and a point cannot be connected to itself. Furthermore, due to our choice of action the connections are \textit{undirected}.

**2.3 The Action**

It is possible to extend the above method to more complicated fields. We choose to stick to the scalar field (the simplest one available) because we are interested in the properties of the discrete space underlying the quantum field theory rather than the theory itself.

Our action of choice (eqn. 2.1) is arguably the simplest one that incorporates both local and neighbour interactions. A notable feature is that all points and connections are \textit{identical}. Note that the parameters $\lambda_2$ and $\xi$ must be positive to ensure that the action is bounded from below.

Optionally, we can add local interaction terms of the general form

\[
\sum_n \frac{\lambda_k}{k!} \phi_n^k \quad k \in 3, 4, 5, \ldots
\] (2.2)

but mostly we will be working with non-interacting theories.
In defining the action we have skipped over a slight ambiguity, namely that we can absorb part of the neighbour interaction terms into the local terms. That is because in an expression like 

$$(\phi_m - \phi_n)^2 = \phi_m^2 + \phi_n^2 - 2\phi_m\phi_n$$

the first two terms on the right hand side are local; they do not mix contributions from different points. We can write

$$S = \sum_n \frac{1}{2} \mu_n \phi_n^2 - \gamma \sum\limits_{\text{neighbours } m,n} \phi_n \phi_m,$$  \hspace{1cm} (2.3)

where \( \mu_n \equiv \lambda_2 + k_n \xi \) and \( \gamma \equiv \xi \). By \( k_n \) we denote the number of neighbours of point \( n \).

We already see that this choice of parameters is a bit unnatural due to the fact that to keep the action bounded from below the parameter \( \gamma/\mu \) is limited to the somewhat artificial range \([0, \frac{1}{k}]\) when \( k_n = k \) is constant.

The ambiguity arises because the two forms of the action are only equivalent when the number of neighbours \( k_n \) is the same at every point. When \( k_n \) differs from point to point we have to choose which one of the two parameters \( \lambda_2 \) and \( \mu \) is constant and which one varies with the number of neighbours. Trial and error has shown us that we should pick the first option. A simple argument will convince the reader that this is indeed the right choice. If we take \( \mu \) to be constant we have to impose the ridiculous demand that \( \gamma/\mu \) be smaller than \( 1/k_{\text{max}} \), where \( k_{\text{max}} \) is the largest number of neighbours of any point in the whole configuration. If we do not, the action is unbounded from below. This would mean for instance that adding one extra point to a square lattice of one billion points (increasing the number of connection from 4 to 5 at some point) would drastically alter the properties of the theory.

It can be helpful to write the action in the form of eqn. 2.3 because that makes it easier to read off the Feynman rules directly. We will only use eqn. 2.3 when concerned with the Feynman rules or when considering regular lattices with a fixed number of neighbours \( k \). In those cases we will express all quantities in terms of \( \gamma \) and \( \mu_n \) rather than \( \lambda_2 \) and \( \xi \).

### 2.4 Finding a Suitable Configuration

All the configurations we can imagine (and additionally satisfy the constraints mentioned above) are allowed in principle but not all of them are useful for our purposes. Here we discuss the minimal requirements a configuration has to satisfy.

First of all, we require **connectedness**. This is because disconnected parts of the configuration (we shall refer to these as **clusters**) are physically completely

---

Note that we rename \( \xi \) as well to distinguish between the two sets of parameters more clearly.
2.4. FINDING A SUITABLE CONFIGURATION

We only need to study connected configurations and we shall automatically know the behaviour of any theory based on a disconnected configuration. To put it in more technical terms: the part of the action corresponding to a cluster is separable and therefore we can integrate the other clusters out.

The second requirement is that the configuration should be extendable to arbitrary size because we want it to have a well-defined continuum limit. If the configuration cannot be extended to larger sizes it will effectively shrink to infinitesimal physical size in the continuum limit.

Furthermore, we want to include an element of randomness in the configurations. We want to do this in such a way that, although the resulting theory can be arbitrarily weird on microscopic scales, it should at some point start to behave in a controlled way. We are particularly interested in configurations that resemble Euclidean spaces on large scales.

**Bethe Lattice**

One type of configuration that we shall use often is the Bethe lattice of coordination number $k$.

The Bethe lattice is a highly symmetric infinite lattice that can be built starting from a single point. The building instructions are

1. Start with a single point
2. Affix $k$ new points to the initial point.
3. Affix $k - 1$ new points to each point added in the last step.
4. Repeat step 1 ad infinitum.

The result of a few of these iterations is shown in figure 2.2. Each point in the lattice has $k$ neighbours and the lattice is translation invariant - it looks exactly the same seen from any point. Note that the Bethe lattice of coordination number 2 is just an infinite line.

The number of points within $r$ steps from the initial point is given by

$$N_r = \begin{cases} 1 + 2r & \text{if } k = 2 \\ 1 + k \frac{(k-1)^r-1}{k-2} & \text{if } k > 2 \end{cases}$$
CHAPTER 2. THEORY

The large number of points makes it hard to study these lattices computationally. For example, a Bethe lattice of coordination number 3 already contains more than three million points within a radius of 20 steps. The rapid (exponential rather than power law) increase of the number of points (volume) as a function of distance for \( k > 2 \) is an indication that the dimension of these configurations is infinite.

The Bethe lattice does not meet our requirements for a useful lattice but it is a nice example of a non-trivial configuration. The simplicity of the lattice allows us to derive analytical results that we can use to cross-check our calculation methods.

**Completely Random Configurations**

In search of configurations that satisfy the ‘minimal requirements’ outlined above we considered different ways to generate configurations with an element of randomness. The initial idea was to make them completely random, in the sense that no macroscopic structure is imposed a priori. This turned out to be an unsatisfactory approach. It is discussed in more detail below.

When trying to generate random configurations, a natural choice is to consider the ensemble of all configurations of size \( N \). After all, there is only a finite number of them. We could posit that every configuration in the ensemble is equally probable and study the properties of configurations randomly drawn from the ensemble.

This method is reminiscent of statistical mechanics where we study an ensemble of microstates and derive expectation values of macroscopic quantities. Even though some of the contributing microstates are very strange, these are so rare that they do not contribute significantly to the macroscopic behaviour. An example is the microcanonical ensemble of a gas in a box: there are microstates in which the gas particles are all on one side of the box while we know that such a thing will never happen in real life. Similarly, even though there are configurations that do not look like a well-behaved space at all, these might be irrelevant in the ensemble of all configurations of size \( N \).

The method meets two of our requirements: it includes an element of randomness and it can be extended to arbitrary sizes by increasing \( N \). The configurations are not necessarily connected but that needn’t be a problem - we can choose to ignore disconnected configurations or to study only the largest connected component.

The problem is that there doesn’t seem to be a well-defined continuum limit because the number of connections per point increases with increasing \( N \). There are \( N(N - 1)/2 \) possible connections between \( N \) points. The number of connections is approximately distributed binomially with a success probability of 0.5 and \( N(N - 1)/2 \) trials. This means that the average number of connections per point is \( (N - 1)/4 \). It is hard to imagine how the behaviour of the theory could stay the same while the number of connections per point keeps increasing. It also looks like the configurations are too compact; for large \( N \) we can reach every point in two steps or less.
The complications described above lead us to abandon this train of thought and look for different ways to generate random configurations.

Percolation Configurations

One class of configurations that meets all the demands - and the principal case that we shall consider - is that of percolation configurations. By taking a regular D-dimensional lattice (I shall refer to these as ‘D-cubic lattices from now on’) and shooting holes in it\(^3\) we get a so-called site percolation configuration. We can also choose to randomly remove connections instead of points; this is called bond percolation.

We call the probability for a point/connection to survive the procedure the survival probability \(p\).

The resulting configurations are likely to be reasonably well-behaved since they are embeddable in a space of limited dimension. Another advantage is that we already have a notion of distance on the lattice, namely the Euclidean distance on the original lattice. Whether this is still the correct notion of distance in the resulting quantum field theory remains to be seen but at least we have something to compare to.

The percolation procedure will not always yield a connected configuration. As we remove more and more points/connections disjoint pieces will start to form. Eventually the configuration will fall apart completely. This is shown for site percolation in figure 2.3. Since we are not interested in disjoint configurations we want to consider only the largest cluster and remove all the smaller ones but we can only reasonably do that when there is a single cluster dominating the configuration. To be precise we need a percolating cluster: a connected part of the configuration that goes from one end of the original D-cubic lattice to the other. Finding out when we have such a cluster is the central question of percolation theory.

It turns out that for infinite percolation lattices there is a critical survival probability \(p_c\) (the percolation threshold) where there is a phase transition of the system. When \(p > p_c\) there is always a percolating cluster and when \(p < p_c\) there is no such cluster. The amazing thing is that the exact value of \(p_c\) has not been calculated for most lattices. Even in the seemingly simple case of two-dimensional site percolation no one has been able to derive the exact value of \(p_c\). Some numerical approximations of percolation thresholds are shown in table 2.1.

### 2.5 Wick Rotation

As the observant reader might have noticed we haven’t mentioned time yet. We are formulating theories with a Euclidean signature. So the spaces we build are actually spacetimes but space and time directions are indistinguishable.

\(^3\)by ‘shooting holes’ I mean removing points at random, also removing the connections associated with the point.
Figure 2.3: Two-dimensional site percolation configurations with varying survival probability $p$. The initial lattice contained 50x50 nodes. Images created using the graph visualization program ‘neato’ from the graphviz package.

<table>
<thead>
<tr>
<th>dimension</th>
<th>site percolation</th>
<th>bond percolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0.5927</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>0.3116</td>
<td>0.24881</td>
</tr>
<tr>
<td>4</td>
<td>0.1969</td>
<td>0.1601</td>
</tr>
</tbody>
</table>

Table 2.1: Approximate percolation thresholds for $D$-cubic lattices.
2.6 Distance

It is not easy to generally define distance on a configuration. A natural candidate is the shortest path length, defined as the number of steps needed to travel from one point to the other. Although intuitively this might seem like a good idea we can easily see that it is not correct. A counterexample on the square lattice is shown in figure 2.4.
What really determines distance in quantum field theory is the propagation of particles. If the propagator from one point to another is larger (compared to another pair of points) then the distance between them is smaller.

Unfortunately, the propagator is not simply linear in the distance. In the lecture notes of the course ‘Theoretical Foundations of Elementary Particle Physics’ by Ronald Kleiss it is shown that in scalar quantum field theory the large-distance behaviour of the propagator on a $D$-cubic lattice is given by

$$\Pi(\tilde{x}) \approx \frac{1}{2} \tilde{m}^{D-2} \left(\frac{2\pi \tilde{m} \tilde{x}}{\lambda_2}\right)^{\frac{D}{2}} \exp\left[-\tilde{m} \tilde{x}\right], \quad (2.4)$$

where $\tilde{x}$ is the Euclidean distance in physical units and $\tilde{m}$ is the mass in units of inverse length.

This means that the propagator decreases approximately exponentially with distance. The scale of the exponential decay is determined by $\tilde{m}$. Note that equation (2.4) is only valid for large distances; at short distances the propagator behaves differently.

We choose to work with the dimensionless distance $x$ in lattice units from now on. It is defined as $x \equiv \tilde{x}/\Delta$ where $\Delta$ is the physical distance between neighbouring points. Similarly we work with a dimensionless mass $m \equiv \tilde{m} \Delta$. Note that the propagator (equation 2.4) only depends on the combination $\tilde{m} \tilde{x} = mx$, apart from an overall scaling. We define the (again dimensionless) Compton wavelength in lattice units $x_C \equiv 1/m$ to emphasize that the dimensionless mass determines the scaling of the theory relative to the lattice. When necessary, we can calculate back and forth between the physical units and lattice units using $\Delta$ which follows from the definition of the parameter $\xi = \Delta^{D-2}$.

The Compton wavelength in lattice units, which determines the scaling of the theory in any dimension, is given by

$$x_C \approx \sqrt{\frac{\xi}{\lambda_2}},$$

as long as $x_C \gg 1$. In all simulations we fix the parameter $\xi$ to one and only vary $\lambda_2$. When presenting results we will often make a remark like ‘We have chosen the Compton wavelength $x_C = 10$’ which means that we have set the parameter values $\xi = 1$ and $\lambda_2 = 0.01$.

The large-distance propagator is not exactly exponential; when $D > 1$ it carries additional factors of $x$. It is of the general form

$$\Pi(x) \propto x^\alpha e^{-x/x_C},$$

where $\alpha$ is some constant. We can rewrite this as

$$\Pi(x) \propto e^{-x/x_C + \alpha \log(x)}.$$
In order to extract the exponential behaviour from the propagator, regardless of dimension, we will study \textit{minus the logarithm} of the propagator. At large distances this quantity

\[- \log (\Pi(x)) = C - \alpha \log(x) + \frac{x}{x_C}\]

becomes linear in the distance with slope \(m = 1/x_C\) in any dimension \(D\).
Chapter 3

Calculation Methods

In the previous chapter we have laid the theoretical foundations for defining a scalar quantum field theory on a general discrete configuration. We have seen some examples, and determined what kind of configurations we wish to study, namely regular $D$-cubic lattices, Bethe lattices and percolation lattices. We are specifically interested in the propagator. Now it is time to see how we can study the configurations.

Firstly, in section 3.1 we review the theoretical framework in which we do calculations. We explain how the Euclidean path integral formalism works and introduce Feynman rules and Green’s functions.

In the next section we use the Feynman rules to derive the propagator on a Bethe lattice. Because a straight line is equivalent to a Bethe lattice of coordination number $k$ this automatically yields the propagator in one dimension.

From section 3.3 on we leave the realm of analytical derivation and start to let the computer do the work for us. We discuss how the propagator on a square lattice can be calculated by numerical evaluation of an integral.

In section 3.4 we show that the propagator calculation using Feynman diagrams can be rephrased in terms of counting paths. We derive an algorithm to count paths on general configurations and show that the propagator can be calculated from its results. The drawback of this path counting method is that it requires $\mu_n$ to be constant. In other words, it only works on configurations with a constant number of neighbours.

We go on to generalize the path counting method to configurations with varying numbers of neighbours in section 3.5. This leads to the improved path counting method which we will shall be using intensively. It is the quickest and most precise method that we have.

Finally we discuss how to get results using random numbers in section 3.6 about Monte Carlo methods. This approach yields less accurate results but it allows us to calculate quantities other than the propagator. It is the only approach that works when an interaction term is added to the action.
CHAPTER 3. CALCULATION METHODS

3.1 Euclidean Path Integral Formalism

We work within the framework of the Euclidean path integral formulation of quantum field theory in which the real action $S$ defines a probability distribution that assigns a probability to each possible state of the field.

A state of the field is fully specified when the value of the field is known at every point. The probability of a particular state $\phi$ ($\phi$ is some function of the labels) is then given by

$$ P[\phi] = \frac{e^{-S[\phi]}}{Z}, $$

where the straight brackets show that $S$ and $P$ are functionals: they map a function to a scalar value.

The normalization factor $Z$ is defined as

$$ Z \equiv \int D\varphi e^{-S[\varphi]}. $$

Note that $\int D\varphi$ is an integral over all possible states $\varphi$. In other words a multiple integral

$$ \int d\varphi_1 \int d\varphi_2 \ldots \int d\varphi_N $$

over the value of the field at each position.

All the quantities we can calculate or measure are expectation values of functions of the field values. For any such functional $f[\phi]$ the expectation value is

$$ \langle f \rangle = \frac{\int D\phi f[\phi] e^{-S[\phi]}}{Z}. \quad (3.1) $$

Below we will discuss several interesting choices for $f$ that we will use to study our systems.

Green’s Functions and Feynman rules

The Green’s functions are defined as the expectation values of products of field values:

$$ G_{n_1, n_2, \ldots, n_N} \equiv \langle \phi_{n_1} \phi_{n_2} \ldots \phi_{n_N} \rangle, $$

where $N$ is the order of the Green’s function. We call a Green’s function of order $N$ an ‘$N$-point Green’s function’.

Measuring the Green’s functions is a fairly straightforward procedure in Monte Carlo simulations. One can also calculate Green’s functions by using Feynman diagrams. One writes down diagrams according to a set of rules that depend on the action of the theory. Each diagram is assigned a mathematical expression in a systematic way and the answer is obtained by summing the contributions of all Feynman diagrams.

The Feynman rules corresponding to the non-interacting action are very simple. The only allowed elements are
3.1. EUCLIDEAN PATH INTEGRAL FORMALISM

- Lines and external lines. These carry an index corresponding to a point label. A line with index \( n \) contributes a factor \( 1/\mu_n \), where \( \mu_n \) depends on the number of neighbours of point \( n \) as explained in section 2.3. A line is called external when one of its ends is not connected to a vertex.

- Two-vertices connecting lines of different indices. This is where the configuration comes into play: two lines with indices \( m \) and \( n \) can only be connected by a two-vertex if the points \( m \) and \( n \) are neighbours. Each two-vertex contributes a factor \( \gamma \).

The mathematical expression corresponding to a diagram in the non-interacting theory is therefore just a product of \( \gamma \)'s and \( 1/\mu_n \)'s.

When we add \( \phi^3 \) and \( \phi^4 \) interaction terms to the action (as defined in equation 2.2) we get two additional Feynman rules:

- Three-vertices connecting three lines with equal index \( n \). These contribute a factor \( -\lambda_3 \)

- Four-vertices connecting four lines with equal index \( n \). These contribute a factor \( -\lambda_4 \).

We will not include any interaction terms of order higher than four.

The \( N \)-point Green’s function \( G_{n_1, n_2, \ldots, n_N} \) is given by the sum of all Feynman diagrams with external lines carrying labels \( n_1, n_2, \ldots, n_N \) and without vacuum bubbles\(^1\). Evaluating the diagrams yields exactly the same results as one would get from evaluating equation 3.1 explicitly.

We can also compute so-called connected Green’s functions \( C_{n_1, n_2, \ldots, n_N} \). These are given by the sum of all connected Feynman diagrams with external lines \( n_1, n_2, \ldots, n_N \).

The most important quantity that we want to calculate is the two-point connected Green’s function \( C_{m,n} \), also known as the propagator.

Properties of the Green’s functions

There are a few properties of the Green’s functions that can be deduced from the Feynman diagrams without too much effort. We state them here for later reference.

- The one-point Green’s function and the one-point connected Green’s function are equal by definition:

\[
C_n = G_n.
\]

\(^1\)A vacuum bubble is a piece of diagram that is not connected to any of the external lines.
• In the non-interacting theory all connected Green’s functions other than
  the two-point functions are zero. In particular each one-point Green’s
  function $G_n$ (the expectation value of the field at position $n$) is zero.

When we only add a $\phi^4$ interaction term to the action the following is true.

• The one-point Green’s functions still vanish.
• $C_{m,n}$ and $G_{m,n}$ are equal. This follows from the above point.
• All odd (connected) Green’s functions vanish.

This means that as long as we don’t include a $\phi^3$ interaction ($\lambda_3 = 0$) the
propagator is simply given by the two-point Green’s function.

### 3.2 Bethe Lattice Propagator

As a first application of the Feynman rules we shall derive the analytical ex-
pression for the propagator on a Bethe lattice.

Please note that the following derivation will later turn out to be wrong. We
tested it against simulations and the resulting expression for the propagator is
off by a constant factor that depends on the Compton wavelength $x_C$ and the
coordination number $k$. We were eventually able to resolve the problem by fixing
the error and using the solution below as an Ansatz. There is undoubtedly a
way to do the derivation correctly from the beginning but we feel that it is more
honest to present the events in the order that they came to pass. Only after
quite some time were we able to track down the error in the derivation. The
solution to the problem is presented in section 5.5.

The propagator on the Bethe lattice must depend exclusively on the shortest
distance $r$ along the lattice - there is no other information due to the high
symmetry. The Schwinger-Dyson equation reads

$$\Pi(r) = \frac{1}{\mu} \delta_0,r + \frac{2}{\mu} \left[(k-1)\Pi(r+1) + \Pi(r-1)\right] \tag{3.2}$$

Note the factor $k - 1$ which reflects the fact that there are multiple steps that
increase the shortest distance while there is only one that decreases the shortest
distance.

We do a Fourier transform

$$R(z) \equiv \sum_r \Pi(r)e^{-irz}$$

which gives us

$$R(z) = \frac{1}{\mu} + \frac{2}{\mu} R(z) \left((k-1)e^{iz} + e^{-iz}\right)$$

$$= \frac{1}{\mu u - \frac{u}{\mu} (1 + (k-1)u^2)}$$
3.2. BETH LATTICE PROPAGATOR

with \( u \equiv e^{iz} \). Transforming back and evaluating the integral we get

\[
\Pi(r) = \frac{1}{2\pi\mu} \int_{-\pi}^{\pi} \frac{u^{r+1}}{u - \mu (1 + (k - 1)u^2)} \, dz
\]

\[
= -i \frac{1}{2\pi\mu} \int_{|u|=1} \frac{u^r}{u - \mu (1 + (k - 1)u^2)} \, du
\]

\[
= i \frac{1}{2\pi\gamma(k-1)} \int_{|u|=1} \frac{u^r}{(u - u_+)(u - u_-)} \, du
\]

where \( u_\pm \) are the solutions for \( u \) of

\[ u - \frac{\gamma}{\mu} (1 + (k - 1)u^2) = 0. \]

They are given by

\[ u_\pm = \frac{\mu}{\gamma} \pm \sqrt{\frac{\mu^2}{\gamma^2} - 4(k-1)} \frac{2(k-1)}{\mu^2} \]

Only \( u_- \) lies within the contour. Therefore, performing the contour integral, we find

\[ \Pi(r) = \frac{1}{\mu} \frac{1}{\sqrt{1 - 4(k-1)\frac{\gamma^2}{\mu^2}}} u_{-r} \]

(3.3)

The propagator is of the form

\[ \Pi(r) \propto e^{-r/r_0} \]

where \( r_0 \) is given by

\[ r_0 \equiv -\frac{1}{\log(u_-)} \]

We write \( u_- \) in terms of \( \lambda_2 \) and \( \xi \)

\[ u_- = \frac{1}{2} \left( \frac{\lambda_2}{\xi} + k - \sqrt{\left(\frac{\lambda_2}{\xi} + k\right)^2 - k^2 + (k - 2)^2} \right) / (k-1). \]

For any \( k, u_- \) is a continuously decreasing function of \( \lambda_2/\xi \) that takes on values between 0 and 1/(k-1). The continuum limit corresponds to taking \( \lambda_2/\xi \rightarrow 0 \).

When the coordination number \( k \) is more than two the following approximation is valid for \( u_- \) close to the continuum limit:

\[ u_- \approx \left(1 - \frac{1}{k - 2} \frac{\lambda_2}{\xi} \right) / (k-1). \]

Note that in this case \( r_0 \) is limited to a maximum value of 1/\( \log(k-1) \).

When the coordination number is equal to two, the approximation becomes

\[ u_- \approx 1 - \sqrt{\frac{\lambda_2}{\xi}} \]


20

CHAPTER 3. CALCULATION METHODS

![Compton Wavelength Graph](image)

Figure 3.1: Compton wavelength in lattice units $x_C \left( \frac{\lambda_2}{\xi} \right)$ on a 1D lattice versus the approximation $x_C \left( \frac{\lambda_2}{\xi} \right) \approx \frac{1}{\sqrt{\lambda_2 \xi}}$. A straight line is shown for reference.

so that

$$r_0 \approx \sqrt{\frac{\xi}{\lambda_2}},$$

is now the familiar $x_C$.

For $k = 2$ the Bethe lattice is just a straight line of points. So as a bonus we have automatically derived the propagator on a 1D lattice. In this special case equation 3.3 reduces to

$$\Pi(x) = \frac{1}{\mu} \frac{1}{\sqrt{1 - 4 \frac{\gamma^2}{\mu^2}}} e^{-x/x_C}. \quad (3.4)$$

The Compton wavelength in lattice units $x_C$ is a complicated function of $\lambda_2/\xi$ but when it is large it is well-approximated as

$$x_C \left( \frac{\lambda_2}{\xi} \right) \approx \frac{1}{\sqrt{\frac{\lambda_2}{\xi}}}. $$

In figures 3.1 and 3.2 it is shown how well the approximation holds. We see that at $x_C = 5$ the approximation is already accurate up to 0.2%. This means that we might as well work with the approximated value as long as the Compton wavelength is more than five.

3.3 2D Propagator by Numerical Evaluation of Integral

We shall now derive another useful result: the propagator on a square lattice.
3.3. 2D PROPAGATOR BY NUMERICAL EVALUATION OF INTEGRAL

Figure 3.2: Difference between \( x_C \left( \frac{1}{\sqrt{\lambda^2 \xi}} \right) \) on a 1D lattice and the approximation \( x_C \left( \frac{1}{\sqrt{\lambda^2 \xi}} \right) \approx 1 / \sqrt{\lambda^2 \xi} \) as a function of the approximated \( x_C \).

In the lecture notes of the course ‘Theoretical Foundations of Elementary Particle Physics’ by Ronald Kleiss it is shown that the propagator on a \( D \)-cubic lattice can be expressed in the form of an integral. In the two-dimensional case we get the integral

\[
\frac{1}{\mu (2\pi)^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} dz_1 dz_2 \frac{e^{i(n_1 z_1 + n_2 z_2)}}{1 - 2\frac{\gamma}{\mu} (\cos z_1 + \cos z_2)}
\]

that can be approximated numerically.

Note that the imaginary part vanishes as it should due to the symmetry of the argument. We therefore only have to evaluate the real part. Secondly, note that the integral is likely to blow up when \( \gamma / \mu \geq 0.25 \).

We can approximate the numerical value as

\[
\Pi(n_1, n_2) \approx \frac{1}{4\mu^2 \left( \frac{2\pi}{N} \right)^2} \sum_{m_1, m_2=0}^{N} \frac{\cos\left( \frac{2\pi}{N} n_1 m_1 + \frac{2\pi}{N} n_2 m_2 \right)}{1 - 2\frac{\gamma}{\mu} \left[ \cos\left( \frac{2\pi}{N} m_1 \right) + \cos\left( \frac{2\pi}{N} m_2 \right) \right]}
\]

where we have divided the intervals \([-\pi, \pi]\) into \( N \) pieces of size \( \Delta \), writing \( z_i = \Delta m_i \) with \( \Delta \equiv \frac{2\pi}{N} \). Note that the above expression is periodic in both arguments with period \( N \) while the real propagator is not. This means that the approximated propagator yields the same answer for \( n_1 = N, n_2 = 0 \) as for \( n_1 = n_2 = 0 \), which is obviously not correct. We therefore expect that we need the sampling rate \( N \) to be much larger than \( n_1 \) and \( n_2 \) to get decent results.
3.4 Path Counting

We now explain the path counting method in which the propagator on any configuration with a constant number of neighbours can be calculated by counting paths.

Remember that the propagator is given by the two-point connected Green’s function which is the sum of all connected Feynman diagrams with two external lines. If \( \lambda_3 = \lambda_4 = 0 \) the only allowed elements in the diagrams are lines and two-vertices. That means that each Feynman diagram contributing to the propagator must simply be a chain of lines connected by two-vertices. These elements contribute a factor \( 1/\mu \) respectively \( \gamma \) to the value of the diagram.

Things are especially simple when the number of neighbours \( k \) is the same for every point in the configuration; \( \mu \) is then position-independent and we can suppress the label \( n \). The total contribution of each Feynman diagram then exclusively depends on the length \( \ell \) of the chain. It is

\[
\frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^\ell.
\]

Remember that the lines in the Feynman diagrams correspond to points in the configuration (they carry a point index) and that a two-vertex can only connect two lines that correspond to neighbouring points. This means that the chain-like Feynman diagrams have a one-to-one correspondence with the paths on the configuration. Each Feynman diagram contributing to \( G_{m,n} \) corresponds to a path between point \( m \) and point \( n \) and, as we have seen above, the contribution of each path depends only on its length.

The propagator between points \( m \) and \( n \) can therefore be expressed as a power series in \( \gamma/\mu \):

\[
\Pi_{m,n} = \frac{1}{\mu} \sum_{\ell=0}^{\infty} N_{\text{paths } m \leftrightarrow n}^\ell \left( \frac{\gamma}{\mu} \right)^\ell,
\]

where the coefficients \( N_{\text{paths } m \leftrightarrow n}^\ell \) are the number of paths between \( m \) and \( n \) of length \( \ell \). If \( \frac{\gamma}{\mu} \) is smaller than its critical value \( 1/k \) the power series converges.

We have now reduced the problem of finding the propagator to one of counting paths. We can do the latter using the computer or by deriving an analytical formula.

**Path Counting Algorithm**

We have developed an algorithm to find the number of paths of each length from an initial point to every other point in the configuration. It is recursive in the path length. Figure 3.3 illustrates how it works.

The algorithm is terminated at some maximum path length \( \ell_{\text{max}} \). A practical problem is that the number of paths increases very quickly with distance. We solved this by storing the logarithm of the number of paths rather than the number of paths itself.
3.4. PATH COUNTING

Figure 3.3: Graphical representation of the path counting algorithm using the square lattice as an example. The red numbers denote the number of paths of length $\ell$ from there to the initial point. The number in each node is the sum of the numbers from the neighbouring nodes in the previous step.

Paths on a D-cubic Lattice

Now that we are on the subject of counting paths we shall also try an analytical approach. We attempt to derive the number of paths on a $D$-cubic lattice. This will provide a useful cross-check for the path counting algorithm.

Consider paths of length $\ell$ between two points with relative position $\vec{x}$. For convenience we flip the axes such that the components of $\vec{x}$ are all positive or zero. We now remark that...

- Each path is fully characterized by a series of steps. One can take steps in $2D$ different directions.
- Considering each axis separately, the total number of steps along axis $i$ is $x_i + 2d_i$, where $d_i$ is the number of detours along that axis. There is a factor two since every detour (basically a step in the wrong direction) must be compensated by an extra step in the right direction.
- This means that along each axis, $x_i + d_i$ steps are taken in one direction and $d_i$ in the other.
- The number of unique ways to order all the steps (and thus define a unique path) is given by
  $$\frac{\left(\sum_{i=1}^{D} x_i + 2d_i\right)!}{\prod_{i=1}^{D} (x_i + d_i)!d_i!}$$
- Any number of detours is allowed as long as the total length of the path is $\ell$. Therefore we add a factor
  $$\delta \left(\ell - \ell_{\text{min}} - 2 \sum_{i} d_i\right),$$
  where $\ell_{\text{min}} \equiv \sum_{i} x_i$ is the shortest possible path length. We see that the number of paths is zero when $\ell < \ell_{\text{min}}$ and when $\ell - \ell_{\text{min}}$ is odd.
Finally, we arrive at the complete formula

\[ N_D(\vec{x}, \ell) = \sum_{d_1 > 0} \sum_{d_2 > 0} \cdots \sum_{d_D > 0} \frac{\ell!}{\Pi_i (x_i + d_i)! (d_i)!} \delta \left( \frac{\ell - \ell_{\text{min}}}{2} - \sum_i d_i \right). \]  

(3.5)

In some cases equation 3.5 allows for easy direct evaluation. For example, for \( D = 1 \) (a line) we find

\[ N_1(x, \ell) = \begin{cases} \ell! \left( \frac{\ell}{2} \right)! \left( \frac{\ell}{2} + x \right)! \left( \frac{\ell}{2} - x \right)! & \text{if } \ell \geq x \text{ and } \ell - x \text{ even} \\ 0 & \text{otherwise.} \end{cases} \]  

(3.6)

For \( D = 2 \) (a square lattice) we can write the self-propagator as

\[ N_2(\vec{0}, \ell) = \begin{cases} \sum_{d=0}^{\ell/2} \frac{\ell!}{(\ell/2 - d)! (d)!^2} & \text{when } \ell \text{ even} \\ 0 & \text{otherwise.} \end{cases} \]  

(3.7)

The non-zero part of the above expression can be simplified to

\[ \frac{4^{\ell/2} \ell! (\ell + 1)!}{\sqrt{\pi (\frac{\ell}{2})^3}} \]

**Paths on a Half-Line**

We can also explicitly calculate the number of paths of length \( \ell \) from the endpoint of a half-line to itself. It is zero when \( \ell \) is odd and \( C_{\ell/2} \) otherwise, where the Catalan numbers \( C_n \) are defined as

\[ C_n \equiv \frac{1}{n + 1} \binom{2n}{n}. \]

We can see this as follows. Imagine the half-line horizontally, running off to infinity on the right. Every path starting from and ending at the endpoint of the half-line is defined by a series of moves to the left (L, toward the endpoint) and to the right (R, away from the endpoint). For example, the sequence

RRRLLL

means ‘go right three steps then go left three steps’. It defines a path of length 6.

Not all sequences are allowed. They have to satisfy two constraints:

1. The number of R’s and L’s must be the same so that the path ends at the endpoint of the half-line.
2. It is not allowed to move left at the endpoint. The sequence RLLR, for example, does not describe a valid path.
3.5. **IMPROVED PATH COUNTING METHOD**

We want to know how many paths there are of a given length that satisfy these constraints.

We can visualize each path on the half-line using a square lattice. We start at \((0,0)\). An \(R\) corresponds to moving one step in the \(x\)-direction and an \(L\) corresponds to moving one step in the \(y\)-direction. This describes a path on the square lattice consisting of steps right (\(r\)) and upwards (\(u\)). Such a path is called **monotonic**. The first of the above constraints is now equivalent to the requirement that a path on the square lattice must end on the diagonal. The second constraint means that a path may not cross the diagonal\(^2\). We now have an equivalent formulation of the problem: what is the number of monotonic paths that do not cross the diagonal on a square lattice of given size?

This is a well-known combinatorial problem. The solution can be found by noting that there is a one-to-one correspondence between monotonic paths that *do* cross the diagonal on an \(n \times n\) lattice and paths on an \((n-1) \times (n+1)\) lattice.

Every diagonal-crossing monotonic path on an \(n \times n\) lattice has a unique point \((x, x+1)\) where it crosses the diagonal for the first time. By reflecting (interchanging \(r \leftrightarrow u\)) the path beyond this point we get a new monotonic path consisting of \(n-1\) moves to the right and \(n+1\) moves upwards.

The reflection procedure is reversible. Every monotonic path in the \((n-1) \times (n+1)\) lattice necessarily crosses the diagonal and by reflecting it in the first point above the diagonal we get a unique monotonic diagonal-crossing path in an \(n \times n\) lattice. Applying the reflection procedure twice does not change the path. This proves the one-to-one correspondence.

We get the number of paths that *do not* cross the diagonal by taking all paths and subtracting from it the number of paths that *do* cross the diagonal. Noting that the number of monotonic paths on an \(m \times n\) lattice is given by \(\binom{m+n}{m} = \binom{m+n}{n}\) we arrive at

\[
N_{\text{paths}}^n = \binom{2n}{n} - \binom{2n}{n+1} = \frac{1}{n+1} \binom{2n}{n} \equiv C_n \quad (3.8)
\]

### 3.5 Improved Path Counting Method

The path counting method only works on configurations with a constant number of neighbours like \(D\)-cubic lattices and Bethe lattices. Unfortunately it is not applicable to percolation lattices. We therefore want to find a generalization of the path counting method.

The problem is that on a general configuration paths of equal length may give different contributions. We remedy this by no longer counting the paths of given

---

\(^2\)It may *touch* the diagonal but not move past it
length individually but only noting their total contribution given the parameters.

Imagine that we want to know the propagator from point $m$ to every other point. We group the Feynman diagrams together by the lengths of the paths they represent. To find the contribution to the propagator to point $n$ of paths of length $\ell + 1$ we take the length-$\ell$ contributions to $n$’s neighbours, add them and multiply by $\gamma/\mu_n$. This is illustrated in fig 3.4. While the algorithm works its way from $\ell = 0$ up to $\ell = \ell_{\text{max}}$ we keep track of the sum of the contributions for each point.

We call this algorithm the improved path counting algorithm (or ‘ipc method’ for short) and it will be our principal tool to determine the propagator.

### 3.6 Monte Carlo Sampling

The final calculation method that we shall consider is called Monte Carlo (MC) sampling. It is based on random numbers.

MC sampling has some advantages over the other available methods. It allows an interaction term to be added to the action and we can use it to calculate much more than just the propagator. Specifically we are now also able to measure the one-point, three-point and four-point Green’s functions and the probability distribution of the field values.

The drawbacks of the MC method are that it demands a lot of computing time and that it yields imprecise results.

#### Theory

Remember that in the Euclidean path integral formalism each state has a probability given by

$$ P[\varphi] \equiv \frac{\int D\varphi e^{-S[\varphi]}}{Z}, $$

with $Z \equiv \int D\varphi' e^{-S[\varphi']}$. All we can calculate are expectation values of functions $f[\varphi]$ of the field values, defined by

$$ < f[\varphi] > \equiv \frac{\int D\varphi f(\varphi)e^{-S[\varphi]}}{Z} $$

Ideally we would iterate over all possible states, calculating $f$ for each and weighing the contribution by the Boltzmann factor $e^{-S}$, sum the contributions and finally divide by the normalization constant $Z$. Unfortunately, the number of possible states is infinite since the field can take on any real value. Even if the field is restricted to a finite set of allowed values the number of states escalates quickly. We can circumvent this problem by means of Monte Carlo sampling. The idea is that a randomly generated finite sample of states may be representative enough to estimate the expectation value of $f$. 
3.6. MONTE CARLO SAMPLING

Figure 3.4: Graphical representation of the improved path counting algorithm using the square lattice as an example. The red expressions denote the total contribution of Feynman diagrams corresponding to paths of length \( \ell \) to the two-point Green’s function from there to the initial point. The expression in each node is the sum of the expressions from the neighbouring nodes in the previous step multiplied by \( \gamma/\mu_n \) where \( \mu_n \) depends on the number of neighbours of the point in question. In this case the \( \mu_n \)’s all have the same value so for aesthetic reasons we have suppressed the labels.
We generate states using a time-homogeneous Markov chain. This means that we will move from state to state in steps and the transition probability to go from the current state $\varphi_a$ to some other state $\varphi_b$ is given by some function $P_T(\varphi_b, \varphi_a)$. The transition probabilities only depend on the current state. This is what is meant by ‘time-homogeneous’. We will refer to moving to the next step as ‘updating’.

We want to define the updating process such that, whatever the initial state is, after updating it many times the probability to end up with a particular state follows the probability distribution given by $P[\varphi]$.

**Metropolis Algorithm**

We will be using the Metropolis algorithm to generate the sample. It satisfies all the requirements of a Markov chain. At each step we edit the current configuration a little and then either accept the change or not based on the change of the action $\Delta S = S_{\text{new}} - S_{\text{old}}$:

$$
P_T = \begin{cases} 
1 & \text{if } \Delta S < 0 \\
e^{-\Delta S} & \text{if } \Delta S \geq 0 
\end{cases}
$$

We edit the configurations by changing the field value at some point by an amount randomly chosen from the range $[-\Delta \varphi, \Delta \varphi]$.

The value of $\Delta \varphi_{\text{max}}$ should be high enough so that the field values can change quickly, but low enough that a large fraction of proposed steps is accepted.

We combine Metropolis steps into sweeps where we loop over all the points once.

**Workflow**

We shall perform all MC simulations according to the following workflow.

- We start out with all field values set to 0.
- We perform several sweeps.
- At regular intervals (the measurement interval) we save the current state. We call this ‘taking a snapshot’.
- The first few snapshots are marked as warm-up measurements. They are still heavily correlated to the initial state and are ignored.
- We stop after taking a predefined number of snapshots.

We call this whole procedure a run. Each run yields a collection of snapshots that we call a sample.

The sample is then used to compute sample averages of the quantities that we are interested in. We simply calculate the quantity for each state and average the result.
The states in a sample are correlated. To avoid biased measurements due to the correlation the MC measurements consist of several runs, each of which is used to compute a sample average. This enables us to compute an estimate $\sigma$ of the error on the results as

$$
\sigma^2 \equiv \frac{1}{N-1} \sum_{i=0}^{N} (x_i - \bar{x})^2,
$$

where $N$ is the number of runs, the $x_i$ denote the sample averages and $\bar{x}$ denotes the average of the $x_i$. Note that $\sigma$ only provides an indication of how stable the results of the MC measurement are, not of how close the results are to the true value. It is very important to keep this in mind when interpreting the results.

For example, if we were to choose a very short measurement interval and warm-up period and a small number of snapshots the results would be very stable (resulting in a small error) but they would be heavily biased towards the initial state. As a rule the estimated error $\sigma$ must be small to get accurate results but the converse is not true: a small $\sigma$ does not guarantee that the results are accurate.
Chapter 4

Method Convergence and Comparison

We are now ready to start using the calculation methods described in the previous chapter. Before gathering conclusive results we realize that all the numerical methods we use are approximations in one way or another. In this chapter we check under which circumstances they yield good and consistent results. We also compare the results obtained by different methods to each other, providing a cross-check.

After we have convinced ourselves of the correctness of each method we move on to the final chapter where we gather and discuss the results.

4.1 Numerical Evaluation of 2D Propagator Integral

The numerical approximation of the two-dimensional propagator integral is a fairly straightforward procedure. The results depend on how finely we sample the integration interval. In figure 4.1 we see that the propagator is only approximated well for distances much smaller than the sample rate. The computing time scales as the square of the sample rate. This limits us to sample rates in the order of 1,000 (i.e. distances up to about 500) for reasonable computation times.

4.2 Path Counting

The path counting method consists of two separate parts. Firstly, given a configuration and an initial point the path counting algorithm returns a list of the number of paths from the initial point to each other point split out by path length. The algorithm is terminated at some maximal path length $\ell_{\text{max}}$. Secondly, this information is used to calculate the propagator.
Figure 4.1: The propagator on a square lattice calculated by numerical evaluation of the integral. The sample rate is the number of subdivisions of the integration interval. We have chosen $x_C = 10$.

The required memory space is usually the limiting factor: it is proportional to the number of points and to $\ell_{\text{max}}$. In most cases the improved path counting method is much more efficient, requiring less memory space and computing time. The path counting algorithm, however, has the advantage that it explicitly shows how the propagator is determined by the number of paths.

Let us compare the analytically derived number of paths (see section 3.4) to those from the algorithm. In table 4.1 we show the number of paths as a function of the path length $\ell$ for four special cases in which we were able to calculate them by hand. In the paragraphs below we describe in detail which formulae were used. In all four cases the path counting algorithm yielded the same results. This strengthens our confidence in the correctness of both the analytical derivations and the algorithm.

To calculate the numbers in table 4.1 we have taken results from section 3.4 and rewritten them in a form that allows recursive computation.

For paths on a line we have used equation 3.6. When $x = 0$ the terms with odd $\ell$ are all zero. For even $\ell$ the number of paths is given by the number of paths of length $\ell - 2$ multiplied by

$$\frac{4}{\ell} \frac{\ell - 1}{\ell}.$$

The $\ell = 0$ term is 1 by definition and all subsequent terms can be calculated recursively. Similarly, when $x = 1$ all the terms with even $\ell$ are zero and the multiplication factor between consecutive non-zero terms is

$$\frac{4}{\ell + 1}.$$
4.2. PATH COUNTING

<table>
<thead>
<tr>
<th>ℓ</th>
<th>line (to self)</th>
<th>line (to nbr)</th>
<th>half-line</th>
<th>square</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>36</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>0</td>
<td>5</td>
<td>400</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>35</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>70</td>
<td>0</td>
<td>14</td>
<td>4900</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>126</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>252</td>
<td>0</td>
<td>42</td>
<td>63504</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>462</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>924</td>
<td>0</td>
<td>132</td>
<td>853776</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>1716</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>3432</td>
<td>0</td>
<td>429</td>
<td>1.17786·10^7</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>6435</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>12870</td>
<td>0</td>
<td>1430</td>
<td>1.65637·10^8</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>24310</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>48620</td>
<td>0</td>
<td>4862</td>
<td>2.3639·10^9</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>92378</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>184756</td>
<td>0</td>
<td>16796</td>
<td>3.41348·10^10</td>
</tr>
</tbody>
</table>

Table 4.1: The number of paths of length ℓ in different situations. From left to right: from a point to itself on a line, from a point to its neighbour on a line, from the endpoint of a half-line to itself and from a point to itself on a square lattice. The results are taken from analytical formulae and are reproduced by the path counting algorithm.
CHAPTER 4. METHOD CONVERGENCE AND COMPARISON

For the number of paths starting from and ending at the endpoint of a half-line we use the Catalan numbers $C_n$ of equation 3.8 as the non-zero terms. These are also related in a simple way:

$$C_n/C_{n-1} = \frac{4n - 2}{n + 1}.$$  

For paths on a square lattice we have used equation 3.7. The consecutive non-zero terms are related by a factor

$$16 \left( \frac{\ell - 1}{\ell} \right)^2.$$  

In all cases we have derived a recursive formula to be able to compute the coefficients of the series without having to evaluate factorials. The recursive formulas (not the original ones) were used to derive the results in table 4.1.

When calculating the propagator the behaviour of the path counting method should be identical to that of its improved counterpart. We will discuss the latter at length in the following next section. In section 4.5 we check that they yield the same results.

### 4.3 Ipc Method

In this section we study the behaviour of the improved path counting method. The results of the improved path counting method depend on the maximal path length $\ell_{\text{max}}$ of the included paths. As paths of greater length are included the calculated value of the propagator increases, converging to its true value. We shall determine how large $\ell_{\text{max}}$ must be to get stable and accurate results by applying the algorithm to $D$-cubic lattices.

Another factor that influences the result is the necessarily finite size of the simulated lattices. We expect to see some boundary effects at distances that are comparable to the lattice size. We shall study how the propagator is altered when the size of the configuration is varied.

We cannot increase the maximal path length and the configuration size indefinitely because the computation time increases accordingly. We shall establish the computational limits of the ipc method, striking a balance between precision and computation time.

In the following four paragraphs we use one, two, three and four-dimensional lattices. In all cases they are periodic $D$-cubic lattices with equal sides of length $s$ so that the total number of points is $s^D$. When referring to ‘distance’ we always mean the shortest Euclidean distance in lattice units. Since the lattices are periodic the longest possible distance is

$$\sqrt{D \left( \frac{s}{2} \right)^2} = \frac{\sqrt{D}}{2} s.$$
Figure 4.2: The propagator on a loop of 500 points calculated using the improved path counting algorithm with varying maximum path length $\ell$. The Compton wavelength $x_C$ is 25. The analytically calculated result is shown for reference.

One-Dimensional Lattice

Let us consider the one-dimensional case first. In figure 4.2 we compare the analytically derived propagator (equation 3.4) to the results from the improved path counting method applied to a large loop configuration. We have varied the maximal path length $\ell$. Note that for low $\ell$ the value of $-\log(\text{propagator})$ is overestimated. This means that the propagator is underestimated just as we had anticipated. As paths of greater length are included, the propagator converges towards the analytically calculated values.

There is also an effect caused by the finite lattice size: the $-\log(\text{propagator})$ lines all bend downwards at the end. We can understand this boundary effect in terms of path counting. As the distance approaches half the loop size, paths going the other way around the loop start to contribute significantly and the propagator becomes larger than it would be on an infinite loop.

From figure 4.2 we see that the ipc method results for the given Compton wavelength ($x_C = 25$) do not change visibly when the maximal path length is increased from 20,000 to 50,000. To be on the safe side we decide that they have certainly converged at $\ell = 50,000$. We have made similar plots using different Compton wavelengths and they show that the required $\ell$ increases with increasing $x_C$. For example, for $x_C = 5$, $\ell = 2,000$ is already sufficient and for $x_C = 50$ full convergence has not been reached even at $\ell = 50,000$. Taking into account a large margin we use the following empirical formula to choose the
minimal value of $l$ when applying the ipc method to a one-dimensional lattice:

$$l \geq 2,000 \times x_C. \quad (4.1)$$

Furthermore, we will only consider distances up to 250, since the results might not have converged yet at larger distances. We will also limit the Compton wavelengths to 25 because otherwise the calculations take too much time.

When the loop is too small the boundary effects start to dominate the propagator. In figure 4.3 we see the results of the ipc algorithm applied to small loop configurations of different sizes. As long as the configuration is large compared to the Compton wavelength the propagator is only affected noticeably near the end. When the configuration is too small (only a few times $x_C$) the whole propagator starts to become affected and we no longer have a good approximation of the propagator on an infinite configuration.

**Two-Dimensional Lattice**

In the two-dimensional case the boundary effects are more pronounced. In figure 4.4 we compare the ipc method results on a large square lattice to those from the numerical evaluation of the propagator integral. Just like in the one-dimensional case there is a dip at the end of the lines but we also see that the lines now start to broaden at about half the lattice size. The latter effect is caused by the fact that the maximal distance is not the same in every direction; in a direction along the lattice it is $s/2$ but on the diagonal it is $s/\sqrt{2}$. So at a given distance some points are already near the 'edge' - resulting in an increased propagator - while others are still far enough from the edge to remain unaffected. We should keep this in mind when considering distances larger than or close to $s/2$. 

Figure 4.3: Propagator from ipc algorithm on loops of various lengths. $x_C = 5$, $\ell = 50,000$. 
4.3. IPC METHOD

Figure 4.4: The propagator on a 500×500 square lattice calculated using the improved path counting algorithm with varying maximum path length $\ell$. The Compton wavelength $x_C$ is 25. The result from numerical evaluation of the two-dimensional propagator integral is shown for reference.
Note that there are now multiple points at a given distance. The propagator is a binned function of the distance and when there are multiple entries in a bin the values are averaged. The averaging is done before taking minus the logarithm.

From the figure we see that the propagator does not change visibly when increasing the maximal path length from 50,000 to 100,000. We therefore determine that for $x_C = 25$ and distances up to 250 the propagator has converged sufficiently at $\ell = 50,000$. Taking into account similar plots for different Compton wavelengths we decide to use the same empirical formula (equation 4.1) as in the one-dimensional case to determine the required $\ell$. In the one-dimensional case we had built in quite a large margin (the propagator had already converged at $\ell = 20,000$) but due to the long calculation times we cannot afford a similar luxury here.

### Three-Dimensional Lattice

In three dimensions we start to experience computational difficulties due to the rapid increase of the number of points with the lattice size. We are practically limited to $s \approx 100$ which corresponds to a largest lattice distance of 86.6. To avoid boundary effects having a large impact on the result we want the Compton wavelength to be much smaller than that, so we will choose $x_C \leq 10$.

An advantage of the smaller distances is that we can suffice with much shorter maximal path lengths. Using the same procedure as before (varying the maximal path length $\ell$ for given $x_C$) we have determined that the results have converged sufficiently at $\ell = 3000$, 5000 and 8000 for $x_C = 5$, 8 and 10 respectively. In three dimensions we will therefore use

$$\ell \geq 800 \times x_C$$

as the rule of thumb to determine the required maximal path length.

In figure 4.5 we have plotted the propagator on three-dimensional lattices of varying sizes to show the influence of boundary effects. We recognize the same broadening of the propagator lines as in the two-dimensional case. We also see that the propagator as a whole is altered significantly due to the finite lattice size. Only the parts of the propagator that do not shift when the lattice size is increased can be assumed to have converged fully. A close-up view of the plot reveals that the $s = 100$ and $s = 120$ propagators approximately overlap up to a distance of 40 for $x_C = 5$ and 30 for $x_C = 10$. Those are the maximal distances for which we can trust the results to resemble those on an infinite lattice.

### Four-Dimensional Lattice

In four dimensions the computational limits are even more stringent. The number of points increases so rapidly with distance that we can only compute very small distances. The best we can reasonably do is a lattice of side $s = 35$ which corresponds to a largest distance of 35. This means that we need to pick

---

1. By ‘the results’ I mean minus the logarithm of the propagator versus the Euclidean distance on a lattice of side $s = 100$. 

---
4.3. **IPC METHOD**

Figure 4.5: Propagator on 3D periodic lattices of varying sizes, calculated using the ipc algorithm. We use two different Compton wavelengths $x_C$ and adjust the maximum path length $\ell$ accordingly.
very small Compton wavelengths and that finite-size effects will still be very dominant. We cannot trust to get a good approximation of an infinite lattice.

Nevertheless we want to extract as much information as possible. Using the same procedure as before we find that for $s = 35$ the results seem to have converged sufficiently at $\ell = 2000$, $2500$ and $3000$ for $x_C = 5$, $6$ and $7$ respectively.

As an illustration of how the lattice size affects the results consider figure 4.6. Only a small part of the propagator (up to a distance of 10) remains unchanged when we increase the lattice size from $s = 30$ to $s = 35$. Only at very short distance can we trust that the propagator has fully converged in the lattice size. Unfortunately this means that the four-dimensional lattice will be of very limited use to us.

4.4 Monte Carlo Sampling

Now that we have successfully explored the possibilities of the ipc method it is time to try out the MC sampling method. Here we consider the behaviour at warm-up and the magnetization. The first results are presented in section 4.5.

Development of the Action

To get a feeling for how quickly the Metropolis algorithm needs to warm up we consider the convergence of the action during the first few sweeps. The action is initially zero because the field values are initialized to zero. It then increases for a while until it reaches a stationary value around which it fluctuates. This
Figure 4.7: Action per point as a function of the number of sweeps for MC sampling on a loop configuration. The length of the loop and the Compton wavelength in lattice units are varied. $\Delta \phi_{\text{max}}$ is set to 1.0.

is shown in figure 4.7. We see that, regardless of $x_C$, the action converges to a final value of approximately $\frac{1}{2}$ per point within a few hundred sweeps. Longer simulations show that the action keeps fluctuating around this value with typical deviations of order 0.01.

Magnetization and Correlation

Another interesting quantity to consider is the magnetization, which is defined as the average of the field values of a given state. In all cases of interest to us the expectation value of the average magnetization is zero. Any deviation from zero is therefore an indication that the MC measurement in question is biased due to correlation within the sample.

As can be seen in figure 4.8 the magnetization can become quite large and changes slowly; if the magnetization is positive in one measurement it will probably still be in the next one. This means that successive measurements are correlated which reduces the effectiveness of the sample. The figure implies that the correlation increases with increasing $x_C$ but does not depend on the number of points.

As an example of how correlation between states in a sample reduces the accuracy of the estimates, consider the one-point Green’s functions or field expectation values

$$G_n = \langle \phi_n \rangle.$$

In section 3.1 we have already remarked that these all vanish in the non-interacting theory. Due to translation invariance the field expectation value
is identical at each point and its estimator from the MC simulation is simply the average magnetization of the sample. From figure 4.8 we immediately see that for large Compton wavelengths the average magnetization is nowhere near zero. For lower $x_C$ the approximation is better because the magnetization is less correlated. The only way to reduce the correlation is to increase the measurement interval.

4.5 Method Comparison

We have now described our main methods. It is important that they yield similar results when applied properly. The possibility of a cross-check between them provides us with an important error-checking tool. We apply all the available methods to $D$-cubic lattices in up to four dimensions.

Let us first consider one-dimensional configurations. For these we have four methods: path counting, ipc, MC and equation 3.4. In figure 4.9 it is shown that the four methods yield compatible results. The ipc and path counting results are exactly the same as those from the analytical formula. The MC method results are compatible with the rest but not nearly as precise. In figure 4.10 we show the same plot with a logarithmic y-axis. We see quite clearly that all results save those from the MC method agree up to very high precision; the MC method results are not accurate enough for this type of plot.

Now let us consider the two-dimensional lattice for which we have the path counting, ipc and MC methods and the numerical evaluation of the propagator
4.5. METHOD COMPARISON

(a) Comparison between the four methods. The results agree so well that the individual lines overlap one another.

(b) A close-up view. We now also show the estimated error $\sigma$ on the MC results. The error computation was described in section 3.6.

Figure 4.9: The propagator on a line configuration calculated using the four different methods. The gray rectangle illustrates the different scaling of the two plots.

The Compton wavelength $x_C$ was 5. The ipc and path counting methods were applied to a loop of size 1,000 using a maximum path length of 50,000. MC sampling was applied to a loop of size 5,000, taking 20 measurements with an interval of 1,000 sweeps after 10,000 warm-up sweeps. $\Delta \phi,_{\text{max}}$ was set to 1. The propagator was averaged over 10 runs.
Figure 4.10: Same as figure 4.9 with the difference that we now plot minus the logarithm of the propagator. The first three lines still overlap exactly. Negative propagator values from the MC data were ignored when computing minus the logarithm. Obviously the MC results are not precise enough for this type of plot.

integral. In figure 4.11 the four methods are compared. Again, all the results are identical but the MC method lacks precision.

In three and four dimensions we only have the ipc, path counting and MC methods at our disposal. These are compared in figures 4.12 and 4.13. Just like in the other cases the methods are in excellent agreement.

4.6 Lattice Effects

Before moving on to the results there are two subtleties that deserve to be mentioned. Both are lattice effects.

Loss of Rotation Invariance at Short Compton Wavelengths

We have seen that the Compton wavelength $x_C$ determines the scale of the theory with respect to the lattice size. In a good approximation of the continuum limit the scale is much larger than the lattice spacing so that the discrete nature of the lattice does not affect the end results. On the other hand, we want to set the scale as small as possible so that we can ‘squeeze’ a large piece of physical space into our finite lattice, enabling us to probe large physical distances. To find the optimal position between these two conflicting requirements we want to know how small $x_C$ can be before lattice effects become apparent.
4.6. LATTICE EFFECTS

Figure 4.11: The propagator on a square grid calculated using four different methods. The gray rectangle in figure (a) shows the plot area of figure (b). The Compton wavelength $\lambda_C$ was 5. The integral was calculated using a sample rate of 1000. The ipc algorithm was applied to a 500×500 periodic square grid using a maximum path length of 50,000. The path counting algorithm was applied to a 100×100 periodic square grid using a maximum path length of 10,000. MC sampling was applied to a 50×50 grid. The propagator was averaged over 20 runs of 20 measurements with an interval of 1,000 sweeps after 10,000 warm-up sweeps. $\Delta_{\phi,\text{max}}$ was set to 1. The error was calculated as described in section 3.6.
Figure 4.12: The propagator on a periodic $20 \times 20 \times 20$ cubic grid calculated using three different methods. The gray rectangle in figure (a) shows the plot area of figure (b). The Compton wavelength $\lambda_C$ was 5. In the ipc algorithm the maximum path length was 5,000. In the path counting algorithm it was 10,000. In the MC sampling the propagator was averaged over 10 runs of 20 measurements with an interval of 1,000 sweeps after 10,000 warm-up sweeps. $\Delta \phi_{\text{max}}$ was set to 1. The error was again calculated as described in section 3.6.
4.6. LATTICE EFFECTS

Figure 4.13: The propagator on a periodic $10 \times 10 \times 10 \times 10$ hypercubic grid calculated using three different methods. The gray rectangle in figure (a) shows the plot area of figure (b). The Compton wavelength $x_C$ was 5. In the ipc algorithm the maximum path length was 5,000. In the path counting algorithm the maximum path length was 10,000. In the MC sampling the propagator was averaged over 10 runs of 10 measurements with an interval of 1,000 sweeps after 10,000 warm-up sweeps. $\Delta \phi_{\text{max}}$ was set to 1. The error was again calculated as described in section 3.6.
Figure 4.14: Propagator on a $200 \times 200$ square lattice for small values of $x_C$. Calculated using the ipc algorithm with path lengths up to 10,000. The propagator as a function of the distance is shown for points positioned on different lines on the lattice. The lines match the formulas shown in the key. Note that the $y$-values in the figure on the top left are truncated because the numerical values become too small for the program to handle.

In one dimension we have already seen that when $x_C$ is much smaller than 5 we can no longer approximate it as $\sqrt{\xi/\lambda}$. This is not that much of a problem; the propagator will always be exponential, regardless of $x_C$. In two dimensions we do encounter some serious problems: when $x_C$ is too low the theory is no longer rotationally invariant. This is shown in figure 4.14. For very short $x_C$ the propagator is no longer a function of the Euclidean distance alone but also of the direction relative to the lattice. At $x_C = 5$ we have regained isotropy. The remaining differences between the lines of different direction are caused by the finite lattice.

We have applied the same procedure to a cubic lattice and this leads to the same result: when $x_C \ll 5$ the propagator is no longer rotationally invariant. Because of these observations we will from now on require that the Compton wavelength $x_C$ be larger than or equal to 5.
4.6. LATTICE EFFECTS

Figure 4.15: Lattice relics visible in a close-up view of the short-distance propagator calculated using the ipc method in up to four dimensions. The Compton wavelength is 5 in all four cases. We see that some of the lines are not smooth, especially around a distance of 2.

In 1D and 2D we used a side $s$ of 500 and a maximal path length of $\ell = 50,000$.
In 3D we used $s = 100$ and $\ell = 6,000$.
In 4D we used $s = 35$ and $\ell = 6,000$.

Irregular Short-Distance Behaviour of Propagators

Another, less influential effect is seen in the short-distance behaviour of the propagators. There are some irregularities there that are likely caused by the lattice. The effect can be seen in figure 4.15 where the lines connecting the data do not always connect smoothly. It is most pronounced in 3D and 4D. These lattice relics are only visible at extremely short distances and do not influence the results significantly. For completeness we note that they are there.
Chapter 5

Results

Now that we know how to use all the calculation methods and have tested them under various circumstances we proceed to gather final results.

In the first section of this chapter we will take a little detour and compute the percolation thresholds. We will see that there is indeed a critical survival probability below which the probability of finding a percolating cluster rapidly drops to zero.

We then study the propagators on $D$-cubic lattices in great detail in order to compare them to propagators on percolation lattices. We then proceed to study propagators on two-dimensional percolation configurations, demonstrating properties such as translation and rotation invariance and comparing them to the regular propagator. We also study propagators on higher-dimensional percolation lattices, but in less detail because we cannot perform very detailed calculations on such lattices.

Another section is devoted to the study of the Bethe lattice. We rederive the analytical expression for the propagator and develop an efficient dedicated algorithm.

Finally we will study the effect of adding a quartic interaction term to the action. We will primarily be concerned with general comparisons between normal and percolation lattices in this respect. Apart from the propagator we will now also study the three- and four-point Green’s functions.

5.1 Percolation Thresholds

First we will study the percolation behaviour of lattices in up to four dimensions, estimating the percolation thresholds.

We estimate the percolation threshold by generating multiple configurations for each survival probability $p$ and plotting the percolating fraction (the fraction of the generated configurations that has a percolating cluster) as a function of $p$. In the algorithm we have defined a percolating cluster as follows. Consider a
CHAPTER 5. RESULTS

Figure 5.1: Percolating fraction for 2D site percolation. One hundred configurations were generated for each value of $p$. The original square lattice was 500×500. For the close-up $p$ was sampled at an interval of 0.001. The horizontal line is at height 0.5; it crosses the data between $p = 0.592$ and $p = 0.593$.

Table 5.1: Estimates of the percolation thresholds from simulation. The quoted numbers are the survival probabilities directly to the left and directly to the right of the point where the percolating fraction crosses 0.5. The percolation thresholds from the literature (as quoted in table 2.1) are shown for reference. The estimates that do not agree with the literature are highlighted in red.

<table>
<thead>
<tr>
<th>percolation type</th>
<th>estimate of $p_C$</th>
<th>$p_C$ from literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D site</td>
<td>0.592 - 0.593</td>
<td>0.5927</td>
</tr>
<tr>
<td>2D bond</td>
<td>0.499 - 0.501</td>
<td>0.5</td>
</tr>
<tr>
<td>3D site</td>
<td>0.3132 - 0.3140</td>
<td>0.3116</td>
</tr>
<tr>
<td>3D bond</td>
<td>0.2496 - 0.2498</td>
<td>0.24881</td>
</tr>
<tr>
<td>4D site</td>
<td>0.197 - 0.198</td>
<td>0.1969</td>
</tr>
<tr>
<td>4D bond</td>
<td>0.160 - 0.161</td>
<td>0.1601</td>
</tr>
</tbody>
</table>

$D$-dimensional percolation lattice formed from an initial lattice of side $d$ where the points have Euclidean coordinates $\vec{x} = (x_0, \ldots, x_{D-1})$. The ‘bottom’ is the set of all points with $x_0 = 0$ and the ‘top’ is the set of all points with $x_0 = d-1$. A cluster is percolating when it has nodes in both the ‘bottom’ and the ‘top’.

The result of this procedure for site and bond percolation in two dimensions is shown in figures [5.1] and [5.2] respectively. We see that there is indeed a percolation threshold where the percolating fraction suddenly changes from zero to one. When zooming in we see that the phase transition is not totally abrupt but is smoothed out a little; this is due to the finite lattice size.

Because the transition between percolating and non-percolating is actually smooth it is impossible for us to determine the exact position of the percolation threshold as it would occur in an infinite lattice. We provide an estimate by trying to pinpoint where the percolating fraction in our finite lattices hits $\frac{1}{2}$. The results are shown in table 5.1. Our estimates are all close to the real values but not all of them are in agreement. Nevertheless, the results are satisfactory given the
5.1. PERCOLATION THRESHOLDS

Figure 5.2: 2D bond percolation. One hundred configurations were generated for each value of \( p \). The original square lattice was 500×500. For the close-up \( p \) was sampled at an interval of 0.001. The horizontal line is at height 0.5; it crosses the data between \( p = 0.499 \) and \( p = 0.501 \).

Figure 5.3: 3D site percolation. One hundred configurations were generated for each value of \( p \). The original cubic lattice was 100×100×100. For the close-up \( p \) was sampled at an interval of \( 4 \times 10^{-4} \). The horizontal line is at height 0.5; it crosses the data between \( p = 0.3132 \) and \( p = 0.3140 \).
CHAPTER 5. RESULTS

Figure 5.4: 3D bond percolation. One hundred configurations were generated for each value of $p$. The original cubic lattice was $100 \times 100 \times 100$. For the close-up $p$ was sampled at an interval of $2 \times 10^{-4}$. The horizontal line is at height 0.5; it crosses the data between $p = 0.2496$ and $p = 0.2498$.

Figure 5.5: 4D site percolation. Twenty configurations were generated for each value of $p$. The original hypercubic lattice has side 50. For the close-up $p$ was sampled at an interval of 0.001. The horizontal line is at height 0.5; it crosses the data between $p = 0.197$ and $p = 0.198$. 
5.2 Bare Propagators on D-cubic Lattices

We now study propagators on normal $D$-cubic lattices in detail. A good understanding of how they behave is crucial to be able to interpret the results on percolation lattices.

### One-Dimensional Lattice

The propagator on an infinite line is given by equation (3.4) which was derived in section 3.2 and confirmed by means of the other methods. In the continuum limit ($x_C \gg 1$) it can be approximated as

$$\Pi(x) \approx \frac{x_C}{2\xi} e^{-x/x_C}.$$  

Note that its behaviour is purely exponential. The Compton wavelength $x_C$ determines the decay length.

There is also an overall factor $\frac{x_C}{2\xi}$ that determines the value of the propagator at $x = 0$. This self-propagator $\Pi(0)$ is by definition the two-point Green’s function from a point to itself. Within the context of the Euclidean path integral formalism this means that

$$\Pi(0) = G_{n,n} = \langle \phi_n^2 \rangle,$$

so that the self-propagator is equivalent to the variance of the field at the given point $n$. In fact, the label $n$ doesn’t matter because the propagator is translation invariant in this case.
CHAPTER 5. RESULTS

Figure 5.7: Distribution of the field values from MC sampling on a loop of 10,000 points for different values of $x_C$. A total of 20 snapshots was taken with 1,000 sweeps between each one, after 10 warm-up measurements. $\Delta \phi_{\text{max}}$ was set to 1.0. The width of the distribution visibly increases with increasing $x_C$. As usual, we keep the parameter $\xi$ at a fixed value of 1.

In the case at hand this tells us that the standard deviation of the field values is given by

$$\sigma_\phi = \sqrt{\frac{x_C}{2\xi}}.$$  

When $x_C$ is increased while $\xi$ is kept constant the field value distribution widens. We can see this directly by studying the field value distribution obtained through MC simulation, see figure 5.7.

When taking the continuum limit ($x_C \to \infty$) the typical field values will become ever larger. If we want to keep them finite we have to compensate by adjusting $\xi$ appropriately.

Two-Dimensional Lattice

Now let us consider the propagator on a square lattice. In figure 5.8 we show the propagator calculated using the ipc method for different values of the Compton wavelength $x_C$. By rescaling the distance axis with $x_C$ we show that the form of the propagator is (almost) unchanged when we vary $x_C$ - its distance dependence is merely rescaled. That is because the large-distance behaviour of the two-dimensional propagator goes as (see equation 2.4)

$$\Pi(x) = \frac{1}{2\sqrt{2\pi}} \sqrt{\frac{x_C}{x}} e^{-x/x_C},$$  

(5.1)
5.2. BARE PROPAGATORS ON D-CUBIC LATTICES

Figure 5.8: Minus the logarithm of the propagator versus the distance in Compton wavelengths for different $x_C$. Calculated using the ipc algorithm on a 500x500 square lattice. Save the usual boundary effects the results are in excellent agreement. Note that the lines are terminated at the same distance in lattice units but at different distances in units of the relevant Compton wavelength. We also show the large-distance propagator calculated from equation 5.1.
which depends on $x$ only in the specific combination $x/x_C$. Note that equation (5.1) only holds at large distances as is demonstrated by the fact that it blows up as $x$ goes to zero.

Upon closer inspection (see figure 5.9) the rescaled propagators do not overlap completely; at short distances they show different behaviour. In particular there is a notable difference in the self-propagator $\Pi(0)$ which shows an $x_C$ dependence.

In figure 5.10 we show how the self-propagator in two-dimensions varies with $x_C$, again keeping $\xi$ at a fixed values of one. The data is approximated very well by a logarithmic fit.

As has been noted before, the self-propagator gives us the standard deviation of the field values. Assuming logarithmic behaviour as in the fit, the standard deviation of the field values is given by

$$\sigma_\phi = \sqrt{a_1 \log(x_C) + b_1}.$$ 

To avoid the field values blowing up in the continuum limit $\xi$ must be adjusted appropriately.

At first glance the large-distance behaviour of $-\log$ (propagator) appears to be very nearly linear. That is to say, the large-distance propagator seems to be approximately exponential. We know from equation (5.1) that the behaviour is
5.2. BARE PROPAGATORS ON D-CUBIC LATTICES

Figure 5.10: The self-propagator as a function of $x_C$ on a 500x500 square lattice. Calculated using the ipc algorithm with maximum path length $2 \times 10^3 x_C$. The data is fitted with a logarithmic and a square root function.

Actually not exactly linear. It is

$$-\log (\Pi(x)) = \log \left(2\sqrt{2\pi}\right) + \frac{1}{2} \log \left(\frac{x}{x_C}\right) + \frac{x}{x_C}.$$  \hfill (5.2)

To see how closely the behaviour approaches linearity and how well the simulated propagator matches equation 5.1 we consider $-\log(\Pi)$ without the linear term $x/x_C$. The result can be seen in figure 5.11. We now get a good indication of the deviation from linearity of minus the logarithm of the propagator. Note that if the propagator was purely exponential the plots would be a horizontal line. Especially the short-distance behaviour of the propagator is not exponential as shown by the steep slope of the lines in this area. Figure 5.11 shows that equation (5.1) doesn’t completely describe the propagator as calculated by the ipc method; this was to be expected. There is a difference at short distances (as the equation is not valid there) and at large distances (due to boundary effects in the ipc method).

The maximum distance that we can probe on a square lattice is limited computationally due to the quadratic increase of the number of points with distance. We can try to avoid this problem by considering rectangular configurations. We choose our configuration to be a long rectangle of length $L$ and width $W$. It is periodic in the sense that the opposite sides of the rectangle are identified.

In figure 5.12 we see the results for a rectangle of length $L = 3,000$, which is six times as large as the side of the square lattice we used previously. We have again plotted minus the logarithm of the propagator with the linear term $x/x_C$ subtracted from it. This helps us to identify very small differences that wouldn’t be visible otherwise. In figure 5.12a we see that the results are not exactly the same but improve as we increase the width of the rectangle.

All in all we can say that the rectangular lattice performs quite well. Remem-
Figure 5.11: The difference between $-\log(\text{propagator})$ and a straight line $x/x_C$ for different values of the Compton wavelength $x_C$. Calculated using the ipc algorithm on a $500 \times 500$ square lattice with maximum path length $2 \times 10^3 x_C$. Also shown is the expected large-distance behaviour given by the non-linear terms on the right-hand side of equation 5.2.

Note that at a distance of 2,000, $-\log(\text{propagator})$ has a value of approximately $2,000/x_C = 400$. With this in mind the observed differences between the rectangular lattice ipc method and equation 5.2 are very small, reaching an accuracy up to the percent level while dramatically decreasing calculation time compared to the square lattice ipc method. However, when we want to study the exact behaviour of the propagator it is not precise enough.

Three-Dimensional Lattice

Now we consider the propagator on a cubic lattice. The large-distance approximation of the propagator is now

$$\Pi(x) = \frac{1}{4\pi \xi x} e^{-x/x_C}, \quad (5.3)$$

which means that

$$-\log \left( \Pi(x) \right) = \log(4\pi \xi x) + \frac{x}{x_C}. \quad (5.4)$$

In figure 5.13 we compare the ipc method results for different Compton wavelengths $x_C$ to equation 5.3. They agree quite well. Just like in the two-dimensional case they differ at large distances due to the ipc method boundary effects and at short distances due to the fact that equation 5.3 is not valid for that regime.

The behaviour of the self-propagator in three dimensions is shown in figure 5.14. Note that it changes very little.
5.2. BARE PROPAGATORS ON D-CUBIC LATTICES

Figure 5.12: The difference between $-\log(\text{propagator})$ and a straight line $x/x_C$ on rectangular lattices of varying width $W$. The length $L$ was 5,000 points in all cases. The calculation was done using the ipc method with varying maximum path lengths $\ell$. The Compton wavelength $x_C$ was chosen to be 5. The large-distance behaviour according to equation 5.2 (without the linear term) is shown for reference.
Figure 5.13: Propagator on a cubic lattice of side 120 for varying Compton wavelengths $x_C$, calculated using the ipc method. The maximum path length $\ell$ was 6,000 for $x_C = 5$ and 8,000 for $x_C = 8$ and $x_C = 10$. We also show the expected large-distance behaviour of the propagator from equation 5.3.

Figure 5.14: Self-propagator as a function of the Compton wavelength $x_C$ on a cubic lattice calculated using the ipc method. The side $s$ of the lattice and the maximal path length $\ell$ are varied to show whether the results have converged. Note that the vertical axis shows an extremely narrow range.
5.2. BARE PROPAGATORS ON D-CUBIC LATTICES

Figure 5.15: Propagator on a hypercubic lattice of side 35 for varying Compton wavelengths $x_C$ calculated using the ipc method. The maximum path length $\ell$ was five, seven and eight thousand for $x_C = 5$, 6 and 7 respectively. We also show the expected large-distance behaviour of the propagator from equation 5.6.

### Four-Dimensional Lattice

In four dimensions it is very hard to get good results from the simulations. The large-distance propagator (see equation 2.4) is

$$\Pi(x) = \frac{1}{4\pi\sqrt{2\pi x_C^2} \xi} \left(\frac{x_C}{x}\right)^{3/2} e^{-mx}.$$  \hfill (5.5)

Equivalently, we can write

$$-\log\left(\Pi(x)\right) = \log\left(4\pi\sqrt{2\pi x_C^2} \xi\right) + \frac{3}{2} \log\left(\frac{x}{x_C}\right) + \frac{x}{x_C}. \hfill (5.6)$$

The behaviour of the propagator calculated using the ipc method is shown in figure 5.15 along with the expected large-distance behaviour according to equation 5.6. The correspondence between the two is quite bad; we can only probe very short distances (where equation 5.6 is not valid) before boundary effects start to become dominant in the ipc method results.

In figure 5.16, the behaviour of the self-propagator in four dimensions is shown as a function of the Compton wavelength $x_C$. It shows only a very slight change.
5.3 Propagators on Two-Dimensional Percolation Configurations

Now that we have intensively studied propagators on regular $D$-cubic lattices we are ready to consider percolation configurations. Particularly we will be concerned with the short-distance and large-distance behaviour of minus the logarithm of the propagator at different values of the Compton wavelength $x_C$ and different survival probabilities $p$. We will also study the self-propagator and check for (approximate) translation and rotation invariance. In this section we restrict ourselves to two-dimensional configurations because these can be studied very precisely using the ipc method. We will study higher-dimensional configurations later on.

Large-Distance Behaviour of the Propagator

When we start to shoot holes in a square lattice the propagator is affected as shown in figure 5.17. As the survival probability $p$ decreases, $-\log(\text{propagator})$ remains approximately linear but its slope increases. This means that the Compton wavelength effectively decreases with decreasing survival probability. The same effect is seen in bond percolation, see figure 5.18.

When the survival probability is too close to the percolation threshold the propagator starts to vary wildly. This is shown in figure 5.19. The irregular and quickly varying results we get in percolation configurations near the percolation...
5.3. PROPAGATORS ON TWO-DIMENSIONAL PERCOLATION CONFIGURATIONS

Figure 5.17: Propagator as a function of the Euclidean distance calculated using the ipc algorithm on randomly generated two-dimensional site percolation configurations with different survival probability $p$. The Compton wavelength $x_C$ is 5. The original lattice has side 500. The propagator is calculated from a single point to all other points and averaged over all points within the same distance bin before taking minus the logarithm. We included paths up to a maximum path length of 10,000.
Figure 5.18: Same as figure 5.17 for bond percolation instead of site percolation.
5.3. PROPAGATORS ON TWO-DIMENSIONAL PERCOLATION CONFIGURATIONS

The short-distance behaviour of the propagator on percolation lattices is shown in figures 5.17b and 5.18b. The large fluctuations caused by the random nature of the lattice make it hard to see how the short-distance behaviour of -log(propagator) changes with the survival fraction. To investigate it further we attempt to reduce the fluctuations by increasing the Compton wavelength $x_C$ and averaging the results over multiple randomly generated configurations. The results are shown in figure 5.21. The fluctuations are now greatly reduced, revealing the survival probability-dependent change in the short-distance propagator.

Rotation Invariance

Earlier (in section 4.6) we saw that on a square lattice the propagator is not rotationally invariant when the Compton wavelength is too small. This tells us that rotation invariance is not a guaranteed property of the discrete quantum field theory - we should check for it.

In figure 5.20 we see that on a two-dimensional percolation configuration – log($\Pi$) is a nice function of the Euclidean distance as long as $p$ is not too close to the percolation threshold. There is only a small spread in the values due to the...
Figure 5.20: 2D histograms showing the correlation between $-\log(\text{propagator})$ and the Euclidean distance in site percolation configurations with different survival probabilities $p$. The values are normalized per horizontal bin, i.e., the sum of the values of each vertical strip is set to 1. Note that the vertical scale is different in each picture.
5.3. PROPAGATORS ON TWO-DIMENSIONAL PERCOLATION CONFIGURATIONS

Figure 5.21: Propagator on $200 \times 200$ site percolation configurations with different survival probabilities $p$ calculated using the ipc method. The Compton wavelength $x_C$ is 25. Paths with lengths up to $2,000 \times x_C$ were included. Before taking minus the logarithm the propagator has been averaged over 10 randomly generated configurations to reduce random fluctuations.

random nature of the configuration. Any rotation invariance would cause the distribution to look like a diverging beam where different slopes in the plot correspond to different directions in the configuration. This effect can be seen in figure 4.14a.

By plotting the propagator as a function of distance for different lattice directions we can check that there is no structural difference. The result is shown in 5.22. As expected, we see no structural difference between the three lines. This leads us to conclude that (approximate) rotation invariance is maintained in percolation lattices when the survival probability is not too close to the percolation threshold.

Translation Invariance

Percolation configurations are not truly translation invariant due to the holes at random positions in the lattice. We do expect approximate translation invariance in the sense that the fluctuations in the propagator become negligible at large distances and that it doesn’t matter which initial point we choose.

To check that this is the case we generate ten configurations and study how much the propagator deviates between them. Additionally, we only consider the propagator along the x-axis so that there is only one point at each distance.

---

1We have applied the same method to bond percolation configurations with $p = 0.7$ and $p = 0.8$. The results are the same.
CHAPTER 5. RESULTS

Figure 5.22: The propagator versus the Euclidean distance on a 200×200 site percolation configuration with a survival fraction \( p = 0.7 \). Calculated using the ipc algorithm with path lengths up to 10,000 and \( x_C = 5 \). The lines represent points whose coordinates obey the equations shown in the labels. In order to reduce fluctuations the propagator was averaged over 10 configurations before taking minus the logarithm.

and random fluctuations are not averaged out among points at the same distance. The result is shown in figure 5.23a where the scale of the fluctuations in the propagator due to the random nature of the configuration is revealed. If the position of the initial point was important we would see that the lines corresponding to some configurations would be systematically above or below the other ones. Instead, we see that all the lines fluctuate around a common center.

At short distances however, see figure 5.23b, the propagator varies quite a bit between individual points; there is a large spread in the value of the self-propagator. This requires a closer look.

Self-Propagator

In figure 5.24 we show the distribution of the self-propagator obtained from a number of randomly generated percolation configurations. The survival probability and the Compton wavelength are varied.

We see from figure 5.24c that the self-propagator distribution widens as the survival probability \( p \) decreases - just as we would expect. Note that at \( p = 1 \) the self-propagator is constant.

The distributions are not smooth but peaked at specific points. In figure 5.24b we have split out the distribution according to the number of neighbours of the point under consideration. From this we see that the immediate surroundings of the point determine the self-propagator to a large extent.
Finally, from figure 5.24 we can deduce that not only the average self-propagator but also the width of its distribution increases with $x_C$. The spread of the self-propagator values is quite large in the sense that the width of the distribution is large compared to its average value.

Intuitively we expect that the self-propagator distribution becomes narrow (compared to its average) in the continuum limit because as we increase the scale the area around each point should start to look approximately the same. To prove this assertion we should map the distribution of the self-propagator at very large Compton wavelengths. At the moment it is not possible for us to do so due to the large computation time required; when we increase $x_C$ we need to include longer paths and use larger configurations.

**Effective Compton Wavelength**

We have seen that the slope of $-\log(\text{propagator})$ on a percolation configuration increases with decreasing survival probability. Here we try to quantify this to some extent.

We fit the $-\log(\text{propagator})$ plots to a linear function using gnuplot. The reciprocal of the slope of the fitted lines is now an approximation of the effective Compton wavelength. The estimated effective Compton wavelength as a function of the survival probability is shown in figure 5.25. Please keep in mind that it is only a rough estimate because the non-linear behaviour of $-\log(\text{propagator})$ is ignored.
CHAPTER 5. RESULTS

(a) The distribution split out by the number of neighbours $k$ of the initial point. $p = 0.7$, $x_C = 5$.

(b) $x_C$ is varied with constant $p = 0.8$.

(c) $p$ is varied with constant $x_C = 5$.

Figure 5.24: Self-propagator distribution on two-dimensional site percolation configurations of size $100 \times 100$ for different $x_C$ and survival probability $p$. Calculated using the ipc algorithm with path lengths up to $2,000 \times x_C$. For each set of parameters ($x_C$ and $p$) 1000 configurations were generated.
5.4. PROPAGATORS ON HIGHER-DIMENSIONAL PERCOLATION LATTICES

Figure 5.25: Estimated effective $x_C$ in site percolation configurations as a function of the survival probability $p$. The estimate is the reciprocal of the slope of a linear fit of $-\log(\Pi)$ as a function of the Euclidean distance. The propagator is calculated using the ipc algorithm with $x_C = 5$ and path lengths up to 10,000 on a 500×500 square lattice.

Figure 5.26: Propagator on cubic site percolation lattices of side $s = 100$ with different survival probabilities $p$. Calculated using the ipc method. The Compton wavelength $x_C$ was 5 and paths of length up to 3,000 were included. The percolation threshold is at $p_c \approx 0.3116$. To the right of the vertical line we expect that boundary effects start to influence the results, see the part on cubic lattices in section 4.3.
CHAPTER 5. RESULTS

Figure 5.27: Same as figure 5.26 but with bond percolation. The percolation threshold is now at $p_c \approx 0.24881$.

Figure 5.28: Propagator on hypercubic site percolation lattices of side $s = 35$ with different survival probabilities $p$. Calculated using the ipc method. The Compton wavelength $x_C$ was 5 and paths of length up to 3,000 were included. The percolation threshold is at $p_c \approx 0.1969$. To the right of the vertical line we expect that boundary effects start to influence the results, see the part on hypercubic lattices in section 4.3.
5.4 Propagators on Higher-dimensional Percolation Lattices

In figures 5.26, 5.27, 5.28 and 5.29 the propagator is shown on cubic and hypercubic percolation lattices. In all cases we see the same effect as before: the \(-\log(\text{propagator})\) lines become steeper at large distances which means that the Compton wavelength is effectively reduced.

5.5 Propagator on a Bethe Lattice

In section 3.2 we derived an analytical formula (equation 3.3) for the propagator on a Bethe lattice. We shall now check whether or not this formula is correct by comparing it to the results from the ipc method.

A computational problem is that the number of points within a given distance increases incredibly fast, especially when the coordination number \(k\) is large. We want to be able to probe large distances and therefore choose \(k = 3\), which is the lowest non-trivial coordination number.

In figure 5.30 the results are shown. To our great surprise we see a discrepancy between the two methods. Aside from the usual finite-lattice effects at the end, the ipc algorithm results are systematically further away. The slope of the line seems to be the same.

\[2\text{Note that in the case of the Bethe lattice we use the shortest path as the measure of distance since there is no Euclidean coordinate system.}\]
Fortunately, we have a third method at our disposal which will be able to discriminate between the two: MC sampling. Even though it is not very precise it will have no problem determining which one of the two predictions is correct by predicting the self-propagator. The results are shown in figure 5.31. We see that the MC method favours the ipc method results and conclude that there seems to be no problem with the ipc algorithm. There are a now a few scenarios left that might explain why the analytical result is different:

1. There is an error in the derivation of the analytical formula. Whatever the error might be, it doesn’t affect the $k = 2$ case; we have checked that rigorously.

2. The finite Bethe lattice is generated wrongly in the code. It is used by both the ipc and the MC method and not by the analytical formula.

3. A finite Bethe lattice is not a good approximation of the infinite Bethe lattice. In this case, all three methods might be correct without contradicting each other.

The problem will be resolved shortly. First we try to improve the performance of the ipc method.

**Dedicated Path Counting Approach to the Bethe Lattice**

Due to the large number of points in a Bethe lattice we run into problems very quickly when trying to compute the propagator using the (improved) path counting method. Fortunately, there is a way to calculate the propagator without having to simulate the entire lattice. Recall that the Bethe lattice is highly

---

**Figure 5.30:** Propagator on a Bethe lattice with coordination number $k = 3$, calculated by applying the ipc algorithm to finite approximations of the lattice with different sizes $L$. Path lengths up to 20,000 were included. The analytical result is also shown.
symmetric; the contribution to the propagator of paths of given length $\ell$ depends only on the distance (shortest number of steps along the lattice) $d$ between the two points. We can use this to our advantage by creating a dedicated version of the ipc algorithm for the Bethe lattice where we group all points together by distance.

Like before, we calculate the contribution to the propagator recursively in the path length. Start from $\ell = 0$ for which by definition there is only one path; it begins and ends at $d = 0$. Its contribution is $\frac{1}{\mu}$. Remember that $\mu$ is constant within a Bethe lattice but does vary among different Bethe lattices as a function of the coordination number $k$. Now we want to get the length $\ell$ contributions given the length $\ell - 1$ contributions. Like before, we imagine extending all the paths of the previous contributions by one step.

Consider paths of length $\ell$ to a point at distance $d$ in a Bethe lattice with coordination number $k$. When the path length $\ell$ is not zero each such path is an extension of a path of length $\ell - 1$. The way in which we can extend the paths depends on where we are in the lattice. There are two cases:

- $d = 0$. Only paths ending at distance 1 can be extended to end at the point at distance 0. There are $k$ points at distance 1 whose paths can be extended.

- $d > 0$. Paths ending at distance $d - 1$ can be extended to the point at distance $d$ and can come from only one point. Paths ending at distance $d + 1$ can also be extended to the point at distance $d$ but can come from
\[ d = 0 \quad d = 1 \quad d = 2 \quad d = 3 \quad d = 4 \]

<table>
<thead>
<tr>
<th>( \ell )</th>
<th>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^0 )</th>
<th>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^1 )</th>
<th>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^2 )</th>
<th>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^3 )</th>
<th>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \ell = 0 )</td>
<td>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^0 \times \frac{\gamma}{\mu} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \ell = 1 )</td>
<td>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^1 \times \frac{\gamma}{\mu} )</td>
<td>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^2 \times \frac{\gamma}{\mu} )</td>
<td>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^3 \times \frac{\gamma}{\mu} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \ell = 2 )</td>
<td>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^2 \times \frac{\gamma}{\mu} )</td>
<td>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^3 \times \frac{\gamma}{\mu} )</td>
<td>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^4 \times \frac{\gamma}{\mu} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \ell = 3 )</td>
<td>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^3 \times \frac{\gamma}{\mu} )</td>
<td>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^4 \times \frac{\gamma}{\mu} )</td>
<td>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^5 \times \frac{\gamma}{\mu} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \ell = 4 )</td>
<td>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^4 \times \frac{\gamma}{\mu} )</td>
<td>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^5 \times \frac{\gamma}{\mu} )</td>
<td>( \frac{1}{\mu} \left( \frac{\gamma}{\mu} \right)^6 \times \frac{\gamma}{\mu} )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.32: Illustration of the dedicated ipc algorithm on a Bethe lattice with coordination number \( k = 3 \). Each row represents the contributions to the propagator of Feynman diagrams corresponding to paths of length \( \ell \). Each row can be calculated easily from the previous one.

\[ k - 1 \] points.

In all cases we take the contribution of all paths ending at distance \( d \pm 1 \) and multiply by \( \gamma/\mu \) to increase the path length by one. Additionally, due to the lattice shape some contributions must be multiplied by a factor \( k \) or \( k - 1 \).

The effect of one step in the recursive formula can be represented as

\[
\begin{pmatrix}
x_0 \\
x_1 \\
x_2 \\
x_3 \\
\vdots
\end{pmatrix}
\rightarrow
\frac{1}{\mu}
\begin{pmatrix}
k x_1 \\
x_0 + (k - 1) x_2 \\
x_1 + (k - 1) x_3 \\
x_2 + (k - 1) x_4 \\
\vdots
\end{pmatrix},
\]

where the \( x_d \) represent the contributions to the propagator corresponding to paths of length \( \ell \) to a generic point at distance \( d \). Applying the transformation yields the contribution of paths of length \( \ell + 1 \).

All we have to do now is to run the recursive algorithm and add the contributions of the different path lengths.

In figure 5.32, we see the result of applying this recursive formula to a Bethe lattice of coordination number \( k \) up to a path length of 4 to further illustrate the principle. In the program the values are calculated numerically rather than algebraically; we fix the value of \( \gamma \) and \( \mu \) beforehand.
5.5. PROPAGATOR ON A BETHE LATTICE

Figure 5.33: Propagators on Bethe lattices of varying coordination number $k$ calculated using the dedicated ipc algorithm. The Compton wavelength $x_C$ was set to 5. We considered distances up to 500 and path lengths up to 50,000.

Using this algorithm we were able to reproduce the one-dimensional propagator and the ipc method results for a Bethe lattice of coordination number $k = 3$. We are now able to easily calculate results for any coordination number $k$ and up to far longer distances; for example, calculating the propagator up to a lattice distance of 500 taking into consideration paths of length up to 50,000 now requires a negligible amount of computing time for $k < 6$. An overview of the results for different $k$ is shown in figure 5.33.

Bethe Lattice Propagator Revisited

In section 3.2 we presented a derivation for the propagator on a Bethe lattice. The results were incompatible with the computer simulations and this lead us to reconsider the derivation. A scrutinious search showed that an error was made right at the beginning in writing down the Schwinger-Dyson equation - see equation 3.2. The problem is that it is only correct for positive $r$. This can be fixed by making the replacement $r \rightarrow |r|$ at two points so that the new equation reads

$$\Pi(r) = 1 + \frac{\gamma}{\mu} (k - 1) \Pi(|r| + 1) + \Pi(|r| - 1) + \delta_{0,r}.$$  \hspace{1cm} (5.7)

Instead of trying to solve this equation from scratch we use the old result as an Ansatz; the simulations already implied that it was correct up to a constant factor. We try a solution of the form

$$\Pi(r) = Cu^{|r|},$$

where $C$ and $u$ are constants that are yet to be determined.
For $|r| > 0$ equation 5.7 reads
\[
\frac{\mu}{\gamma} \Pi(r) = (k - 1) \Pi(|r| + 1) + \Pi(|r| - 1).
\]
Filling in the Ansatz and solving the resulting quadratic equation in $u$ we find
\[
u_{\pm} = \frac{\mu}{2\gamma(k - 1)} \left[ 1 \pm \sqrt{1 - 4\gamma^2(k - 1)/\mu^2} \right],
\]
which is exactly what we found in the old derivation. Based on the old derivation we choose $u = u_-$.

For $r = 0$ equation 5.7 reads
\[
\Pi(0) = \frac{1}{\mu} + \frac{\gamma k}{\mu} \Pi(1).
\]
Filling in the Ansatz and solving for $C$ we find
\[
C = \frac{1}{\mu(k - 1)} \frac{2(k - 1)}{1 + \sqrt{1 - 4\gamma^2(k - 1)/\mu^2}} - 2
\]
The final solution for the propagator on a Bethe lattice of coordination number $k$ is now
\[
\Pi(r) = \frac{1}{\mu k \left(1 + \sqrt{1 - 4\gamma^2(k - 1)/\mu^2}\right)} \nu_{\pm}^{[r]}.
\]
Note that in the case $k = 2$ (propagator on a line) this formula yields the same result as 3.3.

This new and improved equation exactly reproduces the results of the dedicated path counting algorithm.

5.6 Adding an Interaction Term

As a final way to study the behaviour of our lattices we add a $\phi^4$ interaction term to the action. We want to see whether the effect this has on percolation lattices is comparable to the one it has on normal lattices.

We must now use Monte Carlo sampling. Apart from the propagator we will also study the three- and four-point Green’s functions.

Three-Point Green’s function

It has already been remarked in section 3.1 that all odd Green’s functions (like the three-point functions) are zero when the only interaction term is a $\phi^4$ term.

We choose to study the three-point Green’s functions by considering points that lie on a horizontal, vertical or diagonal line and plotting their value as a function of the distance $r$ between the center point and the two outer points.

In figure 5.34 the results are shown for several values of $\lambda_4$. All the results are compatible with the hypothesis that the three-point Green’s functions are zero.
Figure 5.34: The three-point function as a function of distance on a $50 \times 50$ square lattice with $x_C = 5$ for different values of $\lambda_4$. Calculated using MC sampling with 20 runs of 20 measurements. At the start of each run 10 measurements were discarded as a warm-up. The measurement interval was 1000 sweeps. The error is shown in light blue; it was computed as described in section 3.6.
CHAPTER 5. RESULTS

Four-Point Green’s Functions

The four-point Green’s functions are more interesting; they are nonzero even when there is no $\phi^4$ interaction term in the action. The connected four-point Green’s functions are only nonzero when $\lambda_4 \neq 0$.

We can calculate the connected four-point Green’s functions using the measurement data. Because the one-point Green’s functions are zero we can write the four-point Green’s function as

$$G_{n_1,n_2,n_3,n_4} = C_{n_1,n_2,n_3,n_4} + C_{n_1,n_2} C_{n_3,n_4} + C_{n_1,n_3} C_{n_2,n_4} + C_{n_1,n_4} C_{n_2,n_3},$$

where we have explicitly separated it into connected and disconnected parts. As remarked earlier the two-point Green’s functions are equal to their connected counterparts so that we may write

$$C_{n_1,n_2,n_3,n_4} = G_{n_1,n_2,n_3,n_4} - G_{n_1,n_2} G_{n_3,n_4} - G_{n_1,n_3} G_{n_2,n_4} - G_{n_1,n_4} G_{n_2,n_3}.$$  (5.9)

The right-hand side is now fully made up out of quantities that can be measured directly in the Monte Carlo simulations.

We cannot plot the full four-point functions because there are too many degrees of freedom. We therefore restrict ourselves to specific arrangements of points. In particular we will consider sets of four points that are equally spaced along the arc of a circle of radius $r$. There are two easy ways to achieve such an arrangement of points on a square lattice. I will refer to them as ‘+’ configuration and ‘×’ configuration respectively. See figure 5.35. Both are characterised by the radius $r$: the distance of each point to the center of the circle. Note that for an ‘×’ configuration this center need not be a point on the lattice.

Let us assume rotation and translation invariance. Specifically we make the following assumptions:

- The two-point Green’s functions depend only on the distance between the two points. We write

$$G_{n_1,n_2} \equiv G_2(|\vec{x}_1 - \vec{x}_2|),$$

where $\vec{x}_i$ is the Euclidean coordinate of point $n_i$. 

Figure 5.35: Point arrangements that are used when plotting four-point Green’s functions.
5.6. ADDING AN INTERACTION TERM

Figure 5.36: Four-point Green’s function from MC sampling. The total $G_4(r)$ was measured directly from the sample and the estimated error is indicated in light blue. The unconnected contribution was calculated from the two-point functions. The configuration is a 50$x$50 periodic square lattice. We chose $x_C = 5$ and $\lambda_4 = 0$. The MC sampling was done with 20 runs of 10 warm-up measurements followed by 20 real measurements. The measurement interval was 1000 sweeps.

- The four-point (connected) Green’s functions of the ‘+’ and ‘$\times$’ point arrangements depend only on the radius $r$ as defined above. We write them as $G_4(r)$ and $C_4(r)$ regardless of the arrangement in question.

Implementing these assumptions into equation (5.9) we get the following formula for the connected four-point Green’s function as a function of the radius in ‘+’ and ‘$\times$’ point arrangements:

$$
C_4(r) = G_4(r) - 2 \left( G_2(\sqrt{2}r) \right)^2 - \left( G_2(2r) \right)^2. \quad (5.10)
$$

In practical terms this means that if we know the propagator at a distance $d = \sqrt{2}r$ and at a distance $\sqrt{2}d = 2r$ we can calculate the unconnected contribution to the four-point Green’s function at distance $r$. A small complication is that we only know the value of the propagator at lattice points for which $r$ is of the form $\sqrt{m^2 + n^2}$ with integer $m$ and $n$. We don’t want to interpolate the value of the propagator between the measured values. We can therefore only calculate the unconnected contribution to $G_4(r)$ at specific values of $r$ which are different from the values of $r$ at which we can measure $G_4(r)$ itself.

Let us put this somewhat involved procedure to the test. In figure 5.36 we see the four-point Green’s function $G_4(r)$ on a square lattice as measured by the MC method. The unconnected contribution to $G_4(r)$ calculated from the
two-point functions is also shown. The action has $\lambda_4 = 0$ so the two should be identical. Indeed they are.

When we pick a nonzero $\lambda_4$ the connected four-point Green’s function should become different from zero. This means that we expect the measured $G_4(r)$ to deviate from the unconnected part calculated from equation 5.10. In figure 5.37 we see that this indeed happens. Note that even though the four-point Green’s function changes considerably when $\lambda_4$ is increased the connected four-point Green’s function (visible as the deviation between the two lines in the figure) is still very small.

Note that equation 5.10 is not valid for percolation lattices. This is because it requires exact translation invariance and we have already seen that this requirement is not satisfied on percolation lattices; especially the self-propagator shows a large variation with position. Therefore we cannot calculate the connected four-point Green’s function on percolation lattices.

We can still measure the regular four-point Green’s functions. In figure 5.38 we compare them between a square lattice and a square site percolation lattice with a survival probability $p$ of 80%. We also show how setting $\lambda_4$ to one influences the results. We conclude that the four-point Green’s function on a percolation lattice does not appear to be radically different from the one on a normal lattice, apart from some rescaling. Adding an interaction term causes the four-point Green’s function to become smaller by about a factor two in both cases.
5.6. Adding an Interaction Term

Figure 5.38: Four-point Green’s function on a 50×50 square lattice, calculated using MC sampling. We compare a regular lattice to a site percolation lattice with \( p = 0.8 \) at \( \lambda_4 = 0 \) and \( \lambda_4 = 1 \). In the case of the regular lattice the number of runs, number of measurements and measurement interval are 20, 20 and 1000, respectively. With the percolation lattice these parameters were set to 5, 50 and 500, respectively. The estimated errors are shown as light-blue areas around each line.
CHAPTER 5. RESULTS

Now let us consider the effect of the $\phi^4$-interaction on the propagator. Just like we did with the four-point Green’s function we study the propagator on a normal square lattice and a $p = 0.8$ site percolation lattice with both $\lambda_4 = 0$ and $\lambda_4 = 1$. The results are shown in figure 5.39. We conclude again that adding an interaction term has a similar effect on a normal lattice as it does on a percolation lattice.

Please note that we have in no way proven that adding an interaction term has exactly the same effect on regular and percolation lattices. We have simply applied the limited methods that we have available and generally checked that the results were similar in the sense that there were no obvious differences.

---

3If, for example, adding an interaction term actually increased the propagator on a site percolation lattice while decreasing it on a regular lattice we would have seen it in our results.
Chapter 6

Conclusions

We have now reached the end of this Master thesis research. We have come a long way.

Initially we set out to define a framework for quantum field theories on general discrete spaces consisting of points and connections. We sought ways to study them using both analytical and numerical tools. Eventually we decided to focus on percolation lattices and the Bethe lattice; the former because they are non-trivial, random configurations that potentially had a well-defined continuum limit corresponding to $D$-dimensional space, the latter because it was relatively easy to work with.

In the meantime we were able to formulate the calculation of the propagator in terms of the collection of all paths from one point to the other. This path counting method is not only an interesting way to view the propagator but can also be used as an alternative approach to calculate the propagator in simple configurations where the number of paths can be derived analytically, like a line, a half-line or a square lattice.

To efficiently compute the propagator on general discrete lattices we devised a recursive algorithm that we call the improved path counting algorithm, as it originated from trying to generalize the path counting method to configurations with a varying number of neighbours.

As a brief sidestep we have shown by simulation that finite-size percolation configurations indeed show a rapid transition between two regimes where there is almost always or almost never a percolating cluster. Furthermore we have been able to approximately reproduce the values of the percolation thresholds.

We then spent a great deal of time studying the convergence of the results obtained by different methods. We took great care to find the limitations of each method. Furthermore we performed all the cross-checks we could to ensure that both our analytical derivations and our numerical results were correct.

On the Bethe lattice this cross-check lead to a re-evaluation of the analytical formula. Eventually we have derived the propagator on a Bethe lattice of
any coordination number analytically and designed a very efficient dedicated algorithm to compute it numerically.

We then applied the methods to our main interest: \(D\)-dimensional percolation configurations. We have mainly considered the behaviour of the propagator, comparing it to the one on regular lattices. We conclude that in all dimensions (up to four) and for both percolation types the resulting propagator is a well-defined function (save for some small fluctuations) of the Euclidean distance as long as the survival probability is not too close to the percolation threshold. Approximate translation and rotation invariance is maintained. This means that a theory defined on a percolation lattice is expected to have a well-defined continuum limit.

The effect of decreasing the survival probability \(p\) is an overall decrease in the effective Compton wavelength \(x_C\). This means that the physical scale of the theory with respect to the lattice is reduced. The short-distance behaviour of the propagator is also altered, apparently deviating from the normal short-distance behaviour. However, it must be noted that it is unclear how changing the parameters \(\lambda_2\) and \(\xi\) exactly affects the short-distance propagator. Also the exact large-scale behaviour of the propagator (its deviation from pure linearity) cannot be extracted from the simulations due to the lattice fluctuations. Therefore it cannot be ruled out at this point that, in the continuum limit, the propagator on a percolation lattice is equivalent to the one on a normal lattice with different (survival probability-dependent) values of the parameters.

We have measured the self-propagator distribution on two-dimensional percolation lattices and shown how it is largely determined by the direct surroundings of the point in question. We have also shown how the distribution is affected when the Compton wavelength \(x_C\) and the survival probability \(p\) are varied. We conclude that the self-propagator is strongly position-dependent and that there is no sign yet that this can be mitigated by increasing the Compton wavelength.

We have also considered what happens when an interaction term is added to the action. This lead to the conclusion that the propagator and the three-point and four-point Green’s functions show no unexpected behaviour and are affected similarly in regular and percolation lattices.

Of course there is much that could still be done. For instance, as already remarked in section \(5.3\) it would be interesting to see whether or not the self-propagator on a percolation lattice converges to a specific value in the continuum limit. Also it would be interesting to see exactly how the survival probability influences interaction terms by studying the (connected) three-point and four-point Green’s functions as a function of all the relevant parameters. Most importantly it would be very interesting if it could be shown decisively whether or not the continuum limit behaviour of the non-interacting theory in a percolation lattice is equivalent to regular scalar quantum field theory. A helpful tool in this respect would be an analytical derivation of the continuum limit behaviour of the (large-distance) propagator on a percolation lattice.

To conclude, I would like to remark that I have very much enjoyed studying this subject and being able to devote such a large period of time (a full year) to it. It is a rare luxury to be able to tackle a problem in so many different ways, taking an interesting detour whenever one presents itself.
As far as I can think of, the study of quantum field theories on percolation lattices has no foreseeable real-world applications; for me it was simply a matter of curiosity, of trying to solve an interesting puzzle. Nevertheless it was a very useful experience to me and I sincerely hope that my work will be able to aid and inspire others, either through the ideas or through the employed methods.