Universiteit Utrecht

Institute for Theoretical Physics

Master Thesis

Critical Behaviour of the Six Vertex F-model

Author
Niels B. Pannevis

Supervisor
Prof. dr. Gerard T. Barkema

August 20, 2012

Abstract

The F-model is one of the six vertex models, a range of two-dimensional models designed to describe the positions of hydrogen atoms in ice. It exhibits a Kosterlitz-Thouless phase transition, and can easily be mapped onto many other models including the roughening transition in the body centered solid-on-solid model. In this thesis the critical behaviour in the F-model is examined through a computational approach, combined with finite-size scaling and real space renormalization. The simulations were run using two Monte Carlo algorithms; the so-called short loop algorithm, and the full lattice cluster algorithm. The ergodicity of the latter is revealed to be problematic, but this is overcome by combining both algorithms. Also, finite-size scaling is applied, returning a critical temperature of $\epsilon\beta_{\mathrm{c}} = 0.79 \pm 0.02$ compared to the known analytical value of log[2]. Furthermore, although the ice rules prevent local renormalization such as by Kadanoff block spins, a functioning coarse graining prescription based on the height function is proposed. This turns out to renormalise the high temperature states into states with temperatures below the critical point, thus preventing the determination of critical exponents from the renormalization flow.

Contents

Chapter 1

Introduction

Ice is a fascinating material. Not only does it possess peculiar chemical properties, but it also opens a rich world of physics trying to account for this special behaviour.

This thesis focuses on the F-model, that originates in an attempt to describe some of the peculiarities in ice. The F-model, first proposed by Rys in 1963 [1] is an example of the family of six vertex models constructed in the earlier work of Bernal & Fowler [2] and Pauling [3] in their attempt to explain the residual entropy of ice. Lieb solved the F-model analytically in 1967 [4]. Since then, the analytical work on the F-model has been augmented by Lieb & Wu [5], Glasser et al. [6], Brascamp et al.[7], Baxter [8, 9] and others. Also, fruitful simulational approaches have been developed by Rahman & Stillinger [10], Wang et al. [11] and Barkema & Newman [12]. Furthermore, relevant progress has been made on equivalent models, such as the body centered solid-on-solid model constructed by Van Beijeren [13].

The analytical work [4, 14] has shown that the F-model undergoes an infinite-order phase transition of the Kosterlitz-Thouless [15] type. Moreover, it has a large class of equivalent models, and a lattice structure that seems inviting to anyone interested in real space renormalization. This, to our best knowledge, has not been attempted before.

The simulational approach provides an opportunity to apply other than purely mathematical analysis to the F-model. In the research presented here the critical behaviour of the F-model was studied using finite-size scaling and real space renormalization. One of the advantages of this approach is that, unlike the analytical approach, one can possibly also use it to study lattices including defects. However, before getting to that, we first need to develop other than analytical ways to study the F-model on a perfect lattice. The goal of this research was to apply real space renormalization and finite-size scaling to the F-model in order to extract critical coefficients of the phase transition.

The structure of this thesis is as follows.

Chapter 2 sets out the F-model. It starts at the question how to calculate the residual entropy of ice Ih, and from this develops the six vertex models, including the F-model. Following a discussion of the ground states and the basic energetic behaviour of the F-model, two new ways of representing F-model configurations through a height function are introduced. These are then used to discuss several equivalent models that the results obtained in this thesis are relevant to.

The height function approach is employed further in chapter 3 , which deals with the algorithms used to simulate the F-model, i.e. the short loop algorithm and the full lattice cluster algorithm. Their ergodicity and detailed balance are analysed. This includes tackling a previously unknown problem with the ergodicity of the cluster algorithm.

After explaining the F-model and the algorithms employed in the simulations, chapter 4 deals with the first technique used to study the obtained data, i.e. finite-size scaling. It is explained by means of its application to the Ising model, and applied to the F-model. Finite-size scaling produces a remarkable result for the critical temperature of the F-model, for which various possible explanations are discussed in section 4.3.

Chapter 5 concerns the second technique used in this research, that is, real space renormalization. First the general notion behind real space renormalization is set out, which is then illustrated by means of its application to the Ising model. From this it follows that the F-model cannot be renormalized in the usual block-variable approach. However, we develop a different way of coarse graining, and are able to apply real space renormalization to the F-model. The results of this call for some discussion, which is presented in section 5.4.

The main text is concluded by a short summary and some recommendations for further research.

Chapter 2

The F-model

Outline

This chapter sets out the model that is the subject of this thesis. Starting from the molecular structure of ice, the family of six vertex models is constructed and various examples are briefly discussed. After this, the focus is shifted to the F-model and its phase transition. Moreover, the height function is defined, which will be extensively used in later chapters. It is also employed to discuss a class of other models equivalent to square ice or the F-model.

2.1 Towards the F-model

2.1.1 Square Ice

Consider ice. The physics behind this day-to-day material is in fact quite intricate. It has an elaborate phase diagram that contains no less than seventeen different states of ice, although quite a few of these only occur under extreme conditions. Focus on the particular state Ice Ih, the normal[1] state of ice. It has a layered structure of planes that consist of hexagonal cells formed by the oxygen atoms [16]. Each oxygen atom is connected to three oxygen atoms in its own layer and to one oxygen atom above or below it.

Suppose one would like to know the residual entropy S_0 of ice Ih. In Lieb's words [17] residual entropy is "the constant entropy well below the freezing point much larger than that caused by any conceivable lattice vibrations". Important measurements and estimations of this effect were made by Giauque & Ashley [18], and Giauque & Stout [19].

The residual entropy is determined by the number of possible ways to position the hydrogen atoms on the lattice, more specifically their placement on the hydrogen bonds. More about this further below. The number of proton configurations in the ice lattice is for now equal to the partition function Z_0. Taking N to be the number of oxygen atoms and W the average number of configurations per oxygen atom, one would use

$$Z_0 \equiv W^N \qquad\qquad S_0 = N k_B \log(W) \tag{2.1}$$

[1] At pressures between 10 mbar and 1 kbar and temperatures between 173.15°K and 273.15°K water is in the state Ice Ih. Also for many lower temperatures ice takes this form, but the transitions between ice Ih and ice Ic around 173.15°K are quite intricate and beyond the scope of this study. In any case we can conclude that virtually all ice under earthly conditions takes the form Ih.

where k_B is the usual Boltzmann constant. With this, the search for the residual entropy turns into a search for W.

To study the number of proton configurations, consider a two-dimensional square lattice in which the oxygen atoms are put on the vertices, as done earlier by Pauling [3]. See figure 2.1. Each oxygen atom is connected to four other oxygen atoms by hydrogen bonds. The lattice spacing between the oxygen atoms is 2.76 Å [16].

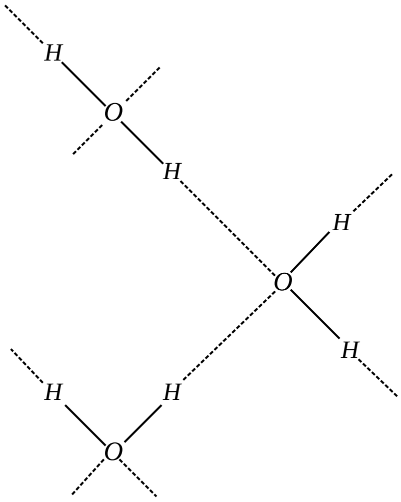


Figure 2.1: *The model used for the molecular lattice of ice Ih.*

The hydrogen-oxygen bond within the original molecule in the gas phase has a length of 0.95 Å. Where Pauling [3] assumed this value to also hold in ice, Hobbs [16] gives an overview of experiments that puts it between 0.95 Å and 1.0 Å . Hence, the proton on the hydrogen bond has only two possible positions: close to one oxygen atom or close to the other.

The fraction of bonds on which two or zero protons are located, the so called 'Bjerrum-defects', is of the order of 10^{-6} [16]. Since the lattices studied in this research only contain up to 40000 bonds (L^2 at $L < 200$), the expected number of Bjerrum defects is zero.

Furthermore the number of ionic defects, places at which an oxygen atom is surrounded by either three protons or by only one, is much lower [16] than the number of Bjerrum-defects. Therefore, we do not account for ionic defects either.

The possible proton configurations are governed by the two ice rules formulated by Bernal and Fowler [2].

1. The proton in a hydrogen bond has two possible positions: close to one oxygen atom or the other.

2. Each oxygen atom has two protons close to it, the two others are closer to it's neighbours.

These rules provide the limits within which to investigate the arrangement of protons on the lattice. In order to do this systematically, put the oxygen atoms on the vertices and denote the relative position of a proton on a bond by an arrow. This results in six possible vertex states, as shown in figure 2.2. All models based on these six states are known as six vertex models.

The simplest six vertex model is that of square ice. In this case the same energy is assigned

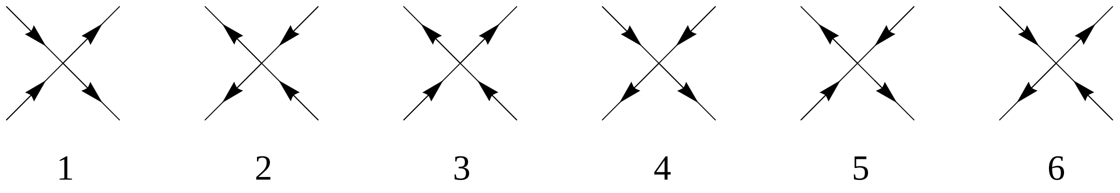


Figure 2.2: *The six vertex states that make up six vertex models.*

to all vertex states. This energy might as well be set to zero. There is no phase transition in the model, it only serves to calculate the residual entropy of square ice. Therefore, the partition function Z is equal to the number of possible proton configurations.

Pauling approximated Z_0 for square ice in a rather crude fashion [3]. At first ignoring the second ice rule, there are 2^{2N} ways to position the protons on $2N$ hydrogen bonds. However, of the now used sixteen possible vertex states only six are allowed. This is corrected for by including a factor $\frac{6}{16}$ for each vertex. Then

$$Z_0 \cong 2^{2N}\left(\frac{6}{16}\right)^N = \left(\frac{3}{2}\right)^N, \qquad \Longrightarrow \qquad W \cong \frac{3}{2}. \tag{2.2}$$

Nagle employed a more elaborate bond graph series expansion approach, which he applied to both square ice [20] and the energetic ice models to be discussed in next section [21]. For square ice Nagle obtained

$$W = 1.540 \pm 0.001. \tag{2.3}$$

Lieb has solved the six vertex model exactly for square ice [17], the F-model [4] and the KDP-model [22], the latter two of which are discussed in the coming paragraph. His analytical solution works on a square lattice with periodic boundary conditions in both directions. It employs the method of transfer matrices, and finds a solution closely related to the Bethe Ansatz for the Heisenberg XXX-spin chain. Sutherland [23] expanded Lieb's solution for square ice much in the same way. The analytical solution yields

$$W = \left(\frac{4}{3}\right)^{\frac{3}{2}} \approx 1.5396... \tag{2.4}$$

This is remarkably close to Nagle's value.

2.1.2 Energetic Ice Models

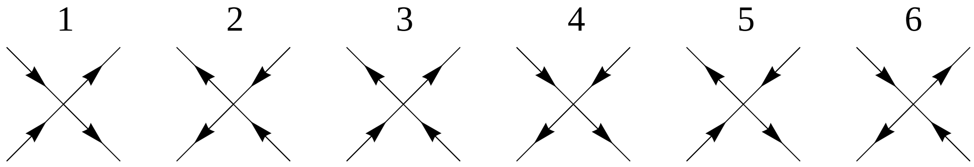
By assigning different energies to different vertex states one can develop more elaborate six vertex models. See figure 2.3.

For all six vertex models the Hamiltonian simply is the sum of the energies of all vertices on the lattice. Denoting the number of vertices in state i by n_i one finds

$$H = \sum_j \varepsilon_j = \sum_{i=1}^{6} n_i \varepsilon_i. \tag{2.5}$$

For specific models the Hamiltonian can be simplified further.

The KDP-model was originally developed by Slater [24] to describe the position of hydrogen



	ε_1	ε_2	ε_3	ε_4	ε_5	ε_6
Square Ice	0	0	0	0	0	0
KDP−model	$-\epsilon$	$-\epsilon$	0	0	0	0
F−model	0	0	0	0	$-\epsilon$	$-\epsilon$

Figure 2.3: *Energy assignment for square ice, the KDP-model and the F-model.*

atoms in potassium dihydrogen phosphate, KH_2PO_4. In this array the phosphate groups play the role of the oxygen atoms in square ice. The vertex states 1 and 2 are assigned a non-zero energy to account for the fact that if the hydrogen atoms are placed as in vertex state 1 or 2 the $(H_2PO_3)^-$ group turns into a dipole. The preferred direction of this dipole reduces the energy of the system. The potassium atoms have fixed positions in the crystal and serve to keep the crystal together, they play no role in the mechanism studied here.

The Hamiltonian of the KDP-model can be simplified to

$$H_{KDP} = -\epsilon(n_1 + n_2). \tag{2.6}$$

As noted earlier, the KDP-model was analytically solved by Lieb [22], who showed that it undergoes a first-order phase transition at

$$\epsilon\beta_{\mathrm{c}} = \log[2], \tag{2.7}$$

where $\beta_{\mathrm{c}} = \frac{1}{k_B T_{\mathrm{c}}}$, and T_{c} is the critical temperature. This was later confirmed by Nagle [25].

A third model, and the subject of this thesis, is the F-model. First proposed by Rys in 1963 [1] it is one of the six vertex models, this time with an energetic preference for the symmetric vertex states 5 and 6. This preference for the symmetric states is not motivated by Rys. He merely proposes it as an interesting model to study since it has somewhat different properties from other models known at the time, such as the Ising model and the Bose Gas.

However, by the right choice of ϵ the F-model can indeed be used to describe a phase transition in ice due to rearrangement of the protons. The negative energy of the symmetric vertex states hereby serves to favour states in which the protons are relatively far away from each other, thus reducing their electromagnetic repulsion. This energy-assignment turns the Hamiltonian of eq. (2.5) into

$$H_F = -\epsilon(n_5 + n_6). \tag{2.8}$$

The F-model was solved analytically through the use of transfer matrices by Lieb in 1967 [4]. With regards to the analytical solution one should also note the accounts of Baxter [8, 9] and Lieb & Wu [5].

Many other six vertex models can be made. Also, by allowing vertices with all four arrows pointing in or out, one can generalize these to so-called eight vertex models, or even further to the sixteen vertex models by also allowing vertices with three arrows pointing in or out of the vertex. For a more elaborate discussion of this, the reader is referred to Lieb and Wu [5]; this thesis focuses on the F-model.

2.2 The Phase Transition in the F-model

2.2.1 Ground States

The preference for vertex states 5 and 6 in the F-model results in two ground states that completely consist of vertex states 5 and 6. See figure 2.4. The two ground states are equivalent, in the sense that they only differ in a relative translation over a single lattice spacing.

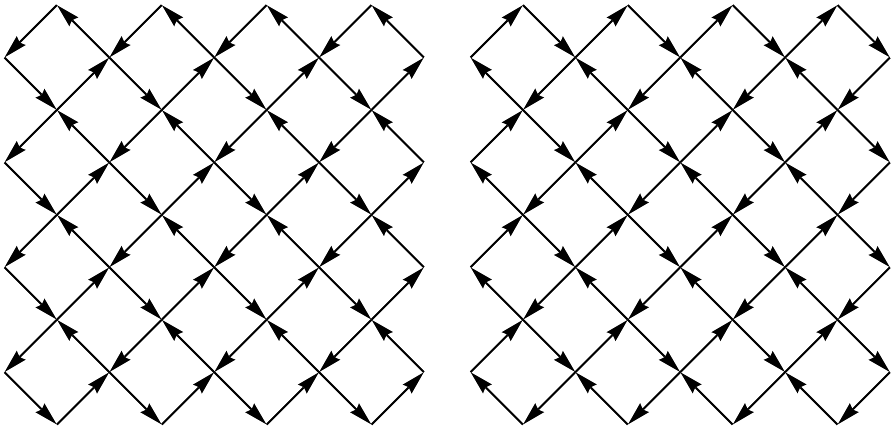


Figure 2.4: *The two ground states for the F-model on a lattice periodic in both directions and of size $L = 8$. On the left, vertices in state 5 occupy odd-numbered vertices, the even vertices are in state 6. On the right, the two are interchanged.*

2.2.2 Rising Temperature

As the temperature rises, the F-model exhibits a phase transition. It moves from the ground state towards a more disordered state in which the symmetric vertex states only cover a fraction of the lattice. Analytically the critical temperature has been determined by Lieb [4] and later verified by Baxter [8] to be

$$\epsilon\beta_{\mathrm{c}} = \log{[2]}. \tag{2.9}$$

The higher the temperature, the more energy there is for the vertices to arrange themselves in other ways than states 5 and 6. See figure 2.5. In these images white and black regions represent the two different ground states, the vertex states 1 through 4 are coloured grey.

At very low temperatures, or very high inverse temperatures $\epsilon\beta = \frac{\epsilon}{k_B T}$, the system is (almost) completely spanned by a single ground state region. Slightly abusing terminology, I refer to the combination $\epsilon\beta$ as inverse temperature, where it technically is the product of the energy of the symmetric vertex states and the inverse temperature. As the temperature rises, or $\epsilon\beta$ lowers, other ground state regions appear. These are at first relatively large, but with rising temperature more and more separate regions are formed. Furthermore, the ground state regions get smaller, and the asymmetric vertex states take up a larger portion of the

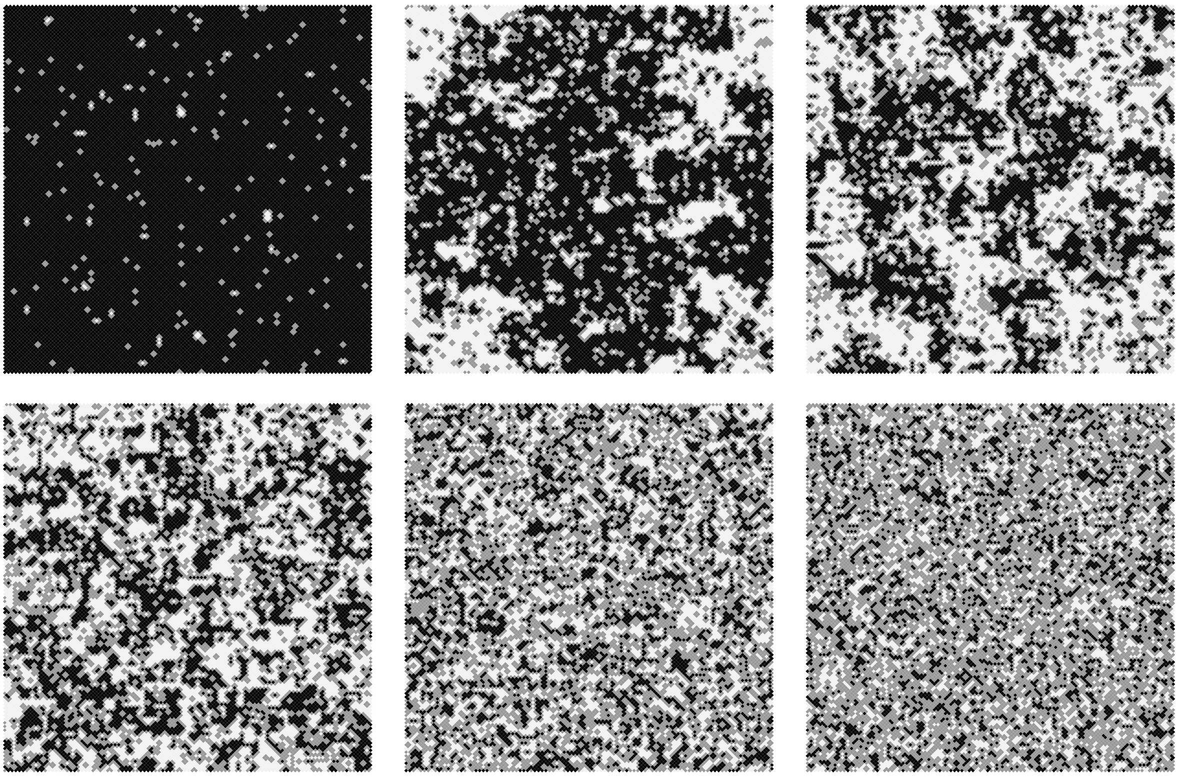


Figure 2.5: *The phase transition in the F-model. Sample configurations for $\beta\epsilon = 1.2$, $\beta\epsilon = 0.8$, $\beta\epsilon = 0.7$, $\beta\epsilon = 0.6$, $\beta\epsilon = 0.3$, $\beta\epsilon = 0$, on a lattice of size $L = 200$. Regions of the two ground states are coloured black and white respectively, the vertex states 1 through 4 are represented by grey squares.*

plane. For very high temperatures the lattice looks chaotic and violent, with many of the vertex states 1 through 4 and only very small ground state regions.

Although a detailed analytical approach is beyond the scope of this project, I would like to quote a few results here in order to shed some light on the nature of the phase transition.

For the F-model, the free energy per site at a temperature below the phase transition ($\epsilon\beta > \epsilon\beta_\mathrm{c}$) is given by [4, 8, 26] [2]

$$f_{T<T_\mathrm{c}}(\lambda) = \epsilon\beta - \frac{\lambda}{2} - \sum_{m=1}^{\infty} \frac{\exp(-m\lambda)\sinh(m\lambda)}{m\cosh(m\lambda)}, \tag{2.10}$$

where $\lambda = \operatorname{arccosh}\left[\frac{1}{2}\exp(2\epsilon\beta) - 1\right]$. For temperatures above the phase transition ($\epsilon\beta < \epsilon\beta_\mathrm{c}$) the free energy has quite a different form

$$f_{T>T_\mathrm{c}}(\mu) = \epsilon\beta - \frac{1}{4\mu}\int_0^\infty \frac{dt}{\cosh(\pi t/2\mu)} \log\left[\frac{\cosh(t) - \cos(2\mu)}{\cosh(t) - 1}\right], \tag{2.11}$$

where $\mu = \arccos\left[\frac{1}{2}\exp(2\epsilon\beta) - 1\right]$.

Raising the temperature towards the critical value the singular part of the free energy

[2] In these analytical studies a slightly different convention was used for the vertex energies: $\epsilon_1 = \epsilon_2 = \epsilon_3 = \epsilon_4 = 1$, $\epsilon_5 = \epsilon_6 = 0$. This has no relevant impact on these results, apart from the fact that they originally do not include ϵ. I have inserted ϵ to show these equations in a uniform convention with the rest of this thesis.

behaves as

$$f_{sing}(t) \sim 4k_B T_c \exp\left(\frac{-\pi^2}{\sqrt{-t}}\right), \tag{2.12}$$

where $t \equiv \frac{T - T_c}{T_c}$. This energy, and all its derivatives go to zero as $t \searrow 0$. Hence this is an infinite-order phase-transition.

Moreover, consider the correlation length ξ. According to the analytical solution it obeys

$$\exp[-1/\xi(\lambda)] = 2x^{1/4} \prod_{m=1}^{\infty} \left(\frac{1 + x^{2m}}{1 + x^{2m-1}}\right)^2 = \prod_{m=1}^{\infty} \left(\frac{1 - y^{2m-1}}{1 + y^{2m}}\right)^2, \tag{2.13}$$

in which $x = \exp(-2\lambda)$ and $y = \exp(-\pi^2/2\lambda)$ with λ as above. The first representation converges quickly for large λ, the latter for small λ.

Near T_c the behaviour is characterized by

$$\xi \sim 4\exp(-\pi^2/2\lambda), \tag{2.14}$$

with in this region $\lambda \sim \sqrt{-t}$. Again, this shows that the phase transition is very different from the usual algebraic divergence. In fact, the F-model undergoes a Kosterlitz-Thouless phase transition [14, 15] much like in the two-dimensional XY model [27].

2.3 The Height Function

The phase transition in the F-model can also be studied through the so-called height function. Based on a specific arrow configuration it prescribes a height for every plaquette or tile on the lattice, i.e. every space between four arrows.

To obtain the heights belonging to an arrow configuration, start at any plaquette on the lattice, and assign a height to this. For choosing this first height, there is one degree of freedom since the arrow configuration only fixes the relative heights. In this thesis the convention is to start at the top left plaquette and assign to it height one, unless stated otherwise.

After fixing this first height, move over the lattice. If, when crossing from one plaquette to the next, the arrow in between is pointing to the right, the next plaquette is assigned a height that is one higher than the previous plaquette. If, however, the arrow in between is pointing to the left the next plaquette will have a height one lower than the previous one. This way the heights for the complete lattice are fixed. Note that no two neighbouring plaquettes can have the same height. The arrow in between always fixes a height difference of one between them.

Consider figure 2.6. Start at the top, defining the height on this position to be 1. Now

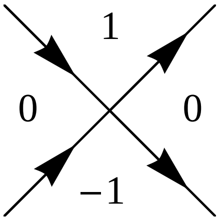


Figure 2.6: *The height function when walking around a single vertex of type one.*

move to the right-hand side of the figure, i.e. clockwise. Since the arrow crossed is pointing to the left relative to the direction of motion, the new height should be one lower, thus 0. Taking another step clockwise, one again passes an arrow that is pointing to the left. Therefore, the bottom plaquette should have height -1. In the following two steps we cross an arrow that is pointing to the right, hence the height raises to 0 and back to 1.

As the above example shows, the ice rules (page 10) ensure that the height function is single-valued. Since any vertex has two arrows pointing in and two pointing out, walking around a vertex one always crosses two arrows pointing to the left, and two pointing to the right. This holds for all vertices so it also holds for the entire lattice. In other words, because there are no divergent vertices the field of arrows is a conservative field, thus the height must be path-independent. For a more rigorous treatment see Appendix 6.2.

In ground state regions the height function is a chequerboard of two values. See figure 2.7.

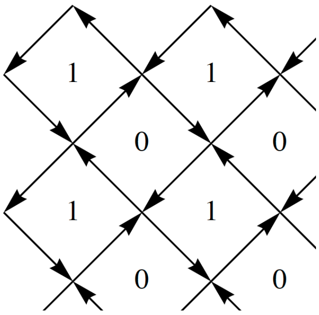


Figure 2.7: *The height function in a ground state region.*

The height function also sheds a new light on the phase transition. First, assign an average height to ground state regions. That is, the configuration in figure 2.7 has a flat height of $\frac{1}{2}$. Now in a plot of the height function the phase transition shows up as a roughening transition. See figure 2.8.

For low temperatures the system is almost completely in the ground state, hence flat. As the temperature rises, there are more and more asymmetric vertex states, and more separate ground state regions that take up a smaller part of the lattice. The asymmetric vertex states incur height differences that make the surface rougher. Above the critical temperature there is no ground state region that wraps around the lattice, i.e. connects to itself through the periodical boundaries. More about this in section 4.2. This also translates to the height correlation function. For $\epsilon\beta > \epsilon\beta_\mathrm{c}$ the surface is in the smooth phase, for $\epsilon\beta < \epsilon\beta_\mathrm{c}$ the surface is rough. The height correlation function then diverges logarithmically [13, 14, 27, 28, 29, 30, 31]:

$$G(r) = \left\langle \left(h(r' \pm r) - h(r')\right)^2 \right\rangle \simeq \frac{2}{\pi \arccos\left[1 - \frac{1}{2}\exp\left(2\beta\epsilon\right)\right]} \log{[r]}. \tag{2.15}$$

Again, this divergence is typical of the Kosterlitz-Thouless phase transition.

2.4 The Three-Colour Representation

Another way of looking at the arrow configurations is through the so called three-colour representation. Its construction is based on the height function. The notion is very simple: take the height function as defined in the last paragraph, but this time take the height modulo

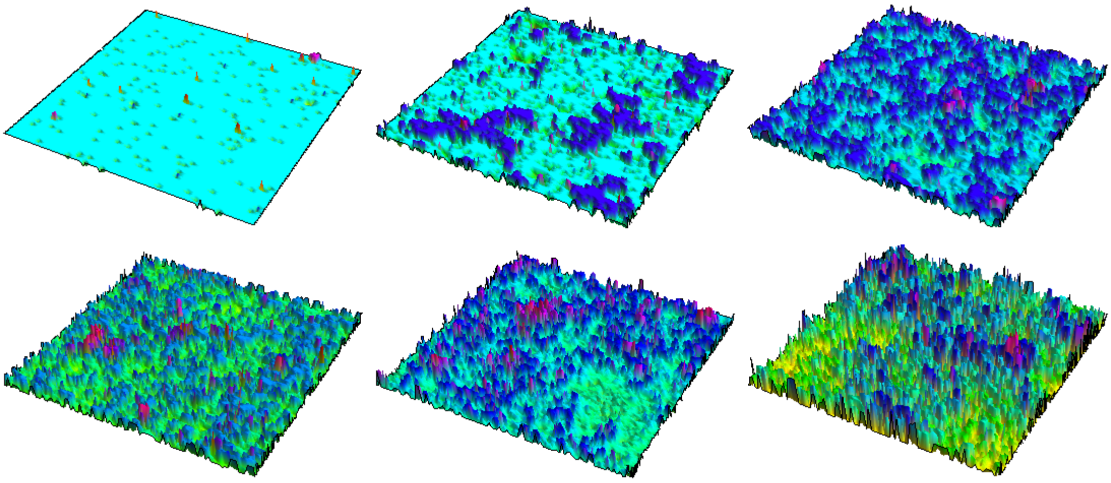


Figure 2.8: *The phase transition in the F-model, as viewed in the height function. Sample configurations for $\epsilon\beta = 1.2$, $\epsilon\beta = 0.8$, $\epsilon\beta = 0.6$, $\epsilon\beta = 0.4$, $\epsilon\beta = 0.2$ and $\epsilon\beta = 0$ on a lattice of size $L = 200$.*

three. Assigning each of these new heights a colour will connect any arrow configuration to a corresponding three-colour representation, or in fact three equivalent ones when we take in the freedom in picking the first plaquette's height, i.e. colour. An example is shown in figure 2.9.

It is important to distinguish this three-colour representation, which is basically the height representation modulo three, from the original height representation. Both will be used in the rest of this thesis, but they are not interchangeable.

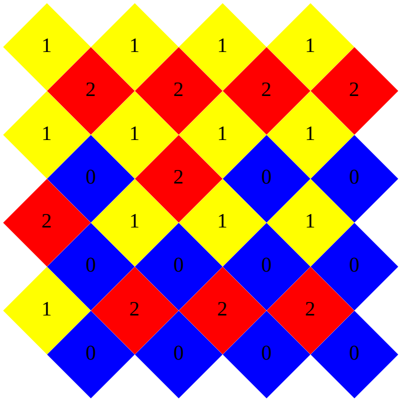


Figure 2.9: *An example of the three-colour representation. The numbers display the value of the height function modulo three.*

2.5 Equivalent Models

The F-model is part of a large class of models that exhibit equivalent behaviour. This means that any results obtained in this thesis will also have relevance for these other models. Note that some equivalences pertain to all six vertex models, others only to the F-model. I do not aspire to cite all known equivalent models.

First of all, square ice can be mapped onto the so-called three-colour problem [5, 32]. The central question to this problem is: in how many ways can one colour a square lattice of given size by using only three colours whilst ensuring that no colour ever borders itself? The three-colour representation discussed in last paragraph gives a mapping from this question to the six vertex models. The only degree of freedom between the colouring and the arrows is picking the colour of the first plaquette. Therefore the number of ways in which the lattice can be coloured is exactly three times the number of allowed configurations in the six vertex models. This is given by equations (2.1) and (2.4).

Secondly, and closely related to this, is the mapping from six vertex models to the anti-ferromagnetic three-state Potts model at temperature zero. Since the configurations of this latter model are formed by lattices of particles which can be in three states but never in the same state as their neighbours, it is essentially the same as the three-colour model. Assigning each particle state a colour turns configurations of the Potts model into configurations in the three-colouring problem and therefore into configurations of six vertex models.

Releasing the modulo three demand and considering the unrestricted height function reveals another model that is closely related to six vertex models. The body centered solid-on-solid model, as constructed by Van Beijeren [13], describes crystal-vacuum interfaces using the height function approach to six vertex models. Various models of surfaces can be constructed by assigning different energies to different vertices, but because of the transition from a flat ground state to a rough surface (see figure 2.8) the F-model is the most interesting of these. In fact this is one of the reasons for this research to be performed on the F-model.

Furthermore, returning to the arrow configurations, one might notice that all vertices are divergence-free. This means that the arrows describe a field with zero curl. Therefore one can consider any arrow configuration to describe the flow of an incompressible fluid [33].

Moreover, quite a few bond graph problems can be shown to be equivalent to questions on six vertex models. Lieb and Wu [5] gave several examples. One of the equivalences rests on the notion that in a bond graph each bond is either activated or not. By comparing two arrow configurations of a six vertex model, and activating those bonds on which the arrows are pointing in the same direction, a relation between bond graphs and six vertex models can be established. This can then be used to solve various questions, for example counting problems surrounding the bond graph configurations. There are several other ways of connecting bond graphs to six vertex models, see for example Lieb & Wu [5], Newman & Barkema [12], and Shugard et al. [27]. Raghavan et al. [34] provides further examples.

Chapter 3

Algorithms

Outline

This chapter treats the two algorithms used to simulate the F-model. Firstly, from two possible loop algorithms the short loop algorithm is shown to be the most effective. This is then treated in more detail. Starting from a basic description, we proceed to the proof of ergodicity and detailed balance.

The second algorithm, the cluster algorithm, is treated along the same lines. The ergodicity of the cluster algorithm turns out to be a more subtle question than it was regarded in previous research. This can be overcome using the short loop algorithm.

3.1 The Short Loop Algorithm

3.1.1 Two Loop Algorithms

To create a Markov chain Monte Carlo algorithm, the first thing needed is a procedure to generate new configurations out of any old configuration. The next step is to try to find a way to thermalize the configurations. Moreover, the algorithm should satisfy ergodicity and detailed balance.

Looking for an algorithm to simulate the F-model, one might propose to flip single arrows one by one. However, flipping a single arrow will create two defects. See figure 3.1(b). At one vertex there are now three incoming arrows (left red circle), at another three outgoing (bottom red circle). As an attempt to repair the vertex with three outgoing arrows one could take one outgoing arrow and flip it. The result is figure 3.1(c). The bottom defect is moved, but there are still two defects. However, we can continue this game. By again flipping one of the outgoing arrows, the defect is moved one position further such that now both defects are on either side of a single arrow, see figure 3.1(d). Flipping this last arrow will cancel both defects at once. The result is figure 3.1(e), which is again an allowed configuration of the F-model, but one that is different from the starting configuration.

This procedure can be generalized. Flipping arrows along any closed loop in which all arrows point in the same direction will always create a new configuration that is also an allowed configuration of the F-model. Since in all vertices along the loop an outgoing arrow is turned into an incoming one, and an incoming arrow is turned into an outgoing one, the ice rules of

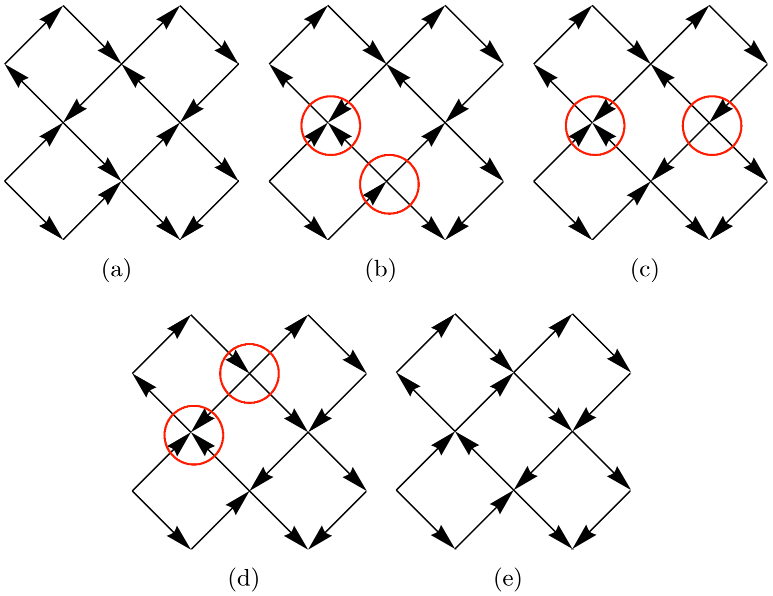


Figure 3.1: *Flipping a single arrow creates two defects, see figure b. By flipping arrows around a closed loop one can again reach an allowed configuration of the F-model. Note these images are cut-outs of a larger lattice, hence they do not satisfy periodicity by themselves.*

section 2.1.1 are still satisfied by the new configuration.

The basis of the loop algorithms is therefore to generate a random closed loop by following arrows on the lattice, and then flipping all these arrows at once. There are two ways to create such a closed loop. One way is to randomly (but always following arrow directions) walk over the lattice until the path ends up at the starting position. The other way is to randomly (but always following arrow directions) walk around the lattice until the path finds itself again, but not necessarily at the starting point. Then only the arrows in the loop are flipped, the arrows in the 'tail' keep their original direction. These two options are known as the long loop and the short loop algorithm. Both algorithms require the lattice to have periodic boundary conditions to ensure that a loop can always be made. The loop algorithms were originally devised by Rahman and Stilinger [10].

The fact that the long loop algorithm takes longer to create a loop does not necessarily mean that it is less effective. Per loop it flips more arrows, which should have its effect on the correlation time. For now it is enough to focus on the short and long loop algorithms for square ice, i.e. without thermalization or a phase transition. This already allows us to compare their efficiency along the lines of Barkema and Newman [12, 35].

The long loop algorithm randomly walks over the two-dimensional lattice until again encountering the starting position. A normal random walk on a square lattice takes a number of steps that scales as L^2 to end up at the starting position. Here L is the number of arrows in one row of the lattice. However, for the long loop algorithm it turns out [12, 35, 36] that the mean length of the loops only grows as $L^{5/3}$. Hence the CPU time per Monte Carlo step also grows as $L^{5/3}$. This does not seem to be good news, but it does not say much about the correlation time. That still needs to be determined.

To measure the correlation time one needs to look at an observable. Inspired by the goal to study the F-model one might consider ρ_{sym}, the density of the symmetric vertex states (5 and 6), that will later be favoured in the F-model. For this observable Barkema and Newman report [12, 35] a correlation time that increases with lattice size as $L^{0.68\pm0.03}$, when measured in the number of Monte Carlo steps. However, the relevant measure is the correlation time when measured as a function of the average number of times each arrow has been turned. Since the mean number of arrows in a loop grows as $L^{5/3}$, the correlation time for the long loop algorithm behaves as

$$\tau_{\text{long loop}} \sim L^{0.68} \frac{L^{5/3}}{L^2} = L^{0.35}. \tag{3.1}$$

The correlation time of the long loop algorithm grows rather slowly as a function of the system size. This is good news, but let us also consider the short loop algorithm.

For the short loop algorithm the correlation time measured in the number of Monte Carlo steps increases much quicker as a function of lattice size. According to Barkema and Newman [12, 35] it grows as $L^{2.00\pm0.01}$. However, the number of arrows in a loop stays constant at about 13.1 arrows. Therefore the correlation time per number of times each arrow has been turned grows much slower as a function of system size:

$$\tau_{\text{short loop}} \sim L^{2.00} \frac{L^{0}}{L^2} = \text{constant}. \tag{3.2}$$

In fact, the correlation time of the short loop algorithm does not increase at all with growing system size, when measured in the number of times each arrow has been turned. This makes the short loop algorithm perform much better for large lattices than the long loop algorithm. Hence we employ the short loop algorithm, and leave the long loop algorithm aside.

3.1.2 Description

The basis of the short loop algorithm was already set out in the last paragraph, here it is considered in more detail. This version of the short loop algorithm is due to Yanagawa and Nagle [37].

The short loop algorithm so far only creates new configurations of square ice out of old configurations of square ice. In order to simulate the F-model and its phase transition the algorithm needs to incorporate the energetic behaviour of the F-model. It should be able to produce a collection of configurations representative of the configurations at a certain temperature. This is done by employing a standard Metropolis scheme. Starting from some configuration μ the algorithm first creates a possible new configuration ν, and then decides whether or not to perform this transition $\mu \to \nu$ based on an acceptance probability. The acceptance probability depends on the energy difference $\Delta E \equiv E_\nu - E_\mu$ between the two configurations.

Since the energy of any configuration is determined by the number of vertices 5 and 6, eq. (2.8), the energy difference ΔE between configuration μ and ν can only depend on the number of vertices 5 and 6 that are created or destroyed in the transition. This only happens along the loop, so this is the only part of the lattice that needs to be accounted for. Thus if m is the number of symmetric vertex states being destroyed in the loop, and n the number of symmetric vertex states being created, then

$$\Delta E = \epsilon(m - n). \tag{3.3}$$

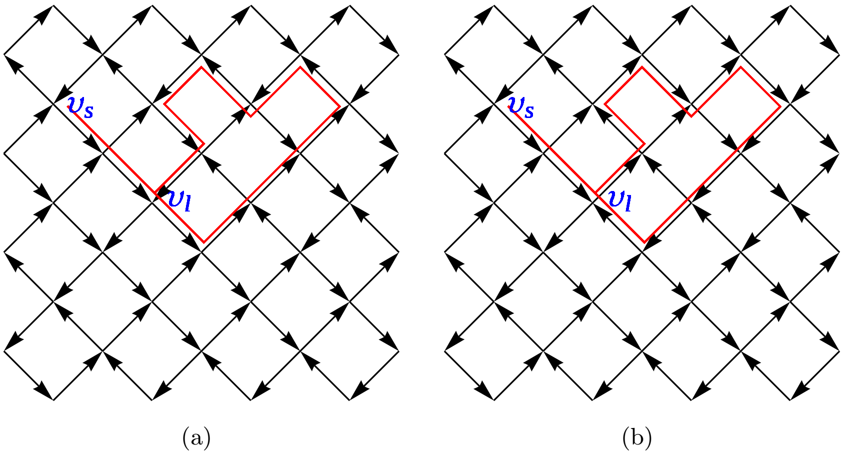


Figure 3.2: *Illustration of the short-loop algorithm.*

We are now in a position to formulate the exact short loop algorithm. It consists of four steps. Also compare figure 3.2.

1. Pick a random vertex (v_s) to start the loop.

2. Make a random walk on the lattice following the arrows. Stop when encountering a vertex that has already been passed through (v_l).

3. Determine the number of symmetric vertex states being created (n) and destroyed (m) by the loop in the path that was just walked.

4. Flip all arrows in the loop with an acceptance probability[3] $A(\Delta E)$:

$$A(\Delta E) = \begin{cases} e^{-\beta \Delta E}, & \text{if } \Delta E > 0 \\ 1, & \text{otherwise.} \end{cases} \tag{3.4}$$

3.1.3 Ergodicity

Given the algorithm set out above, it is now necessary to prove that this algorithm can reach any configuration of the F-model (ergodicity), and that the probabilities for the move from a certain configuration μ to another configuration ν and the reverse move satisfy eq. (3.5) (detailed balance). First, a few words on ergodicity.

With regards to the ergodicity the energetic behaviour is not relevant. What is needed, is to prove that the short loop algorithm can reach any configuration of the F-model, i.e. any configuration that can be created by putting the six vertices of figure 2.2 on a lattice that has periodic boundary conditions.

The six vertex states of figure 2.2 can be turned into one another by flipping two arrows, one outgoing and one incoming. The reader is invited to verify this, it can also be derived from the ice rules, see section 2.1.1.

[3] The advanced reader may note that this is the same acceptance probability as in the Metropolis algorithm for the Ising model. For more information on detailed balance and acceptance ratios the unfamiliar reader is referred to [35, chapter 2].

Consider two random configurations of the F-model and draw bonds on the line segments in which the arrows are pointing in different directions. At any vertex an even number of lines will meet. This can only be so if the lines form (possibly intersecting) loops on the lattice. These loops may wrap around the lattice, using the periodic boundary conditions.

Now notice that the allowed vertices can only be changed into other allowed vertices by flipping an equal number of in- and outgoing arrows. Also, the arrow on any bond can only have two directions, so that if it is not the same as in the original configuration it must be pointing the other direction. Therefore, all these loops formed by bonds on which the arrows are pointing in a different direction than in the original lattice must in fact be loops along which all arrows point in the same direction. Hence, one can reproduce the original configuration by flipping all the arrows on such a loop in a step of the short loop algorithm.

It may not be in a single Monte Carlo step, but every configuration of the F-model can be reached from any other configuration by means of the short loop algorithm. Hence the short loop algorithm satisfies ergodicity.

3.1.4 Detailed Balance

Any algorithm employed to simulate a physical system should not be biased towards a certain type of configurations, other than the distribution that the configurations are physically in. To ensure correct sampling by the algorithm, one should consider the fraction $\frac{P(\mu\to\nu)}{P(\nu\to\mu)}$, where $P(\mu \to \nu)$ is the probability of moving from configuration μ to configuration ν. As expressed by Barkema and Newman [35, page 39], the algorithm will sample the configurations according to the Boltzmann distribution as long as

$$\frac{P(\mu \to \nu)}{P(\nu \to \mu)} = e^{-\beta(E_\nu - E_\mu)}. \tag{3.5}$$

This equation is referred to as detailed balance. Technically this term has a slightly different meaning, but in this context the relevant demand made by detailed balance is the above equation.

$P(\mu \to \nu)$ can be broken down further. For $g(\mu \to \nu)$ the probability of generating a move that will take the system from configuration μ to configuration ν, and the acceptance probability $A(\mu \to \nu)$ as described above, eq. (3.5) becomes

$$\frac{P(\mu \to \nu)}{P(\nu \to \mu)} = \frac{g(\mu \to \nu)}{g(\nu \to \mu)} \frac{A(\mu \to \nu)}{A(\nu \to \mu)} = e^{-\beta(E_\nu - E_\mu)}. \tag{3.6}$$

Evaluate this equation piece by piece.

Firstly focus on the part $\frac{g(\mu\to\nu)}{g(\nu\to\mu)}$. Consider a move that starts at vertex v_s, then walks over a path $P = \{v_s, ..., v_l\}$ before making the loop $\{v_l, ..., v_m\}$ in which the arrows are flipped. For the loop to be a loop we need $v_l = v_m$. Again, see figure 3.2. The probability of this process is the product of the probabilities of the three different steps.

The probability of picking the starting point v_s is $\frac{1}{N}$, where N is the number of vertices. This is followed by a walk along l vertices, with at each vertex two possible options. Hence, the path before the loop contributes a factor 2^{-l}. The loop is a path along $m - l$ vertices. The probability of making this loop is $2^{-(m-l)}$. In total

$$g(\mu \to \nu) = \frac{1}{N} 2^{-l} 2^{-(m-l)} = \frac{1}{N} \left(\frac{1}{2}\right)^m. \tag{3.7}$$

Now consider the reverse transition, $g(\nu \to \mu)$. Again, there is a probability of $\frac{1}{N}$ of picking v_s as the starting point. Then one walks along the same path of l vertices, again with probability 2^{-l}. The loop has now been flipped, so to get the original configuration μ back one should walk along the loop in the inverse direction. This has the same probability $2^{-(m-l)}$. Hence, in the same way as before

$$g(\nu \to \mu) = \frac{1}{N} 2^{-l} 2^{-(m-l)} = \frac{1}{N}\left(\frac{1}{2}\right)^{m}. \tag{3.8}$$

Therefore,

$$\frac{g(\mu \to \nu)}{g(\nu \to \mu)} = \frac{\frac{1}{N}2^{-m}}{\frac{1}{N}2^{-m}} = 1. \tag{3.9}$$

The $g(\mu \to \nu)$ is worked out here for one specific path. Technically the tail, the part of the path that is not in the loop in which the arrows are flipped, does not contribute to the change $\mu \to \nu$. This change is fully determined by the loop. Taking all possible paths that generate the same transition $\mu \to \nu$ together, one would sum over all possible tails and all possible places for the tail to connect to the loop, keeping the loop fixed. However, this summation would have the same effect on $g(\mu \to \nu)$ as it has on $g(\nu \to \mu)$. Therefore their ratio would not change.

Now turn to the acceptance ratios defined by eq. (3.4). It is easy to see that the short loop algorithm will satisfy eq. (3.6). If the configuration ν has a higher energy than μ, i.e. $\Delta E \equiv E_\nu - E_\mu > 0$ then $A(\mu \to \nu) = e^{-\beta(E_\nu - E_\mu)}$, whilst $A(\nu \to \mu) = 1$. If it is the other way around, and configuration μ has an energy above that of configuration ν, then $\Delta E \equiv E_\nu - E_\mu < 0$ and $A(\mu \to \nu) = 1$, whilst $A(\nu \to \mu) = e^{\beta(E_\nu - E_\mu)}$. So in any case

$$\frac{A(\mu \to \nu)}{A(\nu \to \mu)} = e^{-\beta(E_\nu - E_\mu)}. \tag{3.10}$$

Combined with eq. (3.9) this shows that eq. (3.6) and therefore eq.(3.5) and detailed balance are satisfied.

Hence the short loop algorithm satisfies both detailed balance and ergodicity. It correctly samples the physical configurations of the the F-model. For more elaborate simulations of ice using the short loop algorithm see Barkema and De Boer [38].

For our purposes, the short loop algorithm may not always be the most efficient algorithm. Even though the correlation times have an ideal dynamical exponent of zero (eq. (3.2)) the short loop algorithm is still quite inefficient. This is caused by the low acceptance probability. In fact, Barkema and Newman report [12, 35] an acceptance probability of only 36% at the critical temperature so that almost two thirds of the time the computer is doing work that will be thrown away. One could try to improve this algorithm by adjusting the walk over the lattice so that it is not random but favours those vertices where it can create a symmetric vertex out of an asymmetric one, but adding such a preference whilst still satisfying ergodicity and detailed balance is quite a tedious task. Therefore we turn our attention to another algorithm for the F-model.

3.2 The Cluster Algorithm

3.2.1 Description

The second algorithm employed in this project is the cluster algorithm. This algorithm is based on the three-colour representation discussed in section 2.4. Since the three-colour representation of the six vertex models is closely related to the $q = 3$ anti-ferromagnetic Potts model, it is not surprising that the cluster algorithm was originally developed while Wang, Swendsen and Kotecký [11] were studying the latter model. In this research the refined version of Barkema and Newman [12, 35] is used. For simplicity, I first present the core steps and then expand to the full algorithm. Ergodicity and detailed balance are then proven directly for the full algorithm.

The cluster algorithm begins with the translation of the arrow configuration into a lattice of three colours. On this coloured lattice it proceeds to build clusters, much in the same way the Wolff algorithm operates for the Ising model. Summarized in six steps, the basis of the cluster algorithm is as follows.

1. Translate the arrow configuration into a three-colour lattice.
2. Pick a plaquette, call its colour A.
3. Randomly pick another colour $B \neq A$.
4. Build a cluster:
   (a) Add all neighbouring plaquettes of colour A or B.
   (b) Continue until all possible additions have been considered.
5. Interchange the colours $A \rightleftarrows B$.
6. Translate the three-colour lattice back to an arrow configuration.

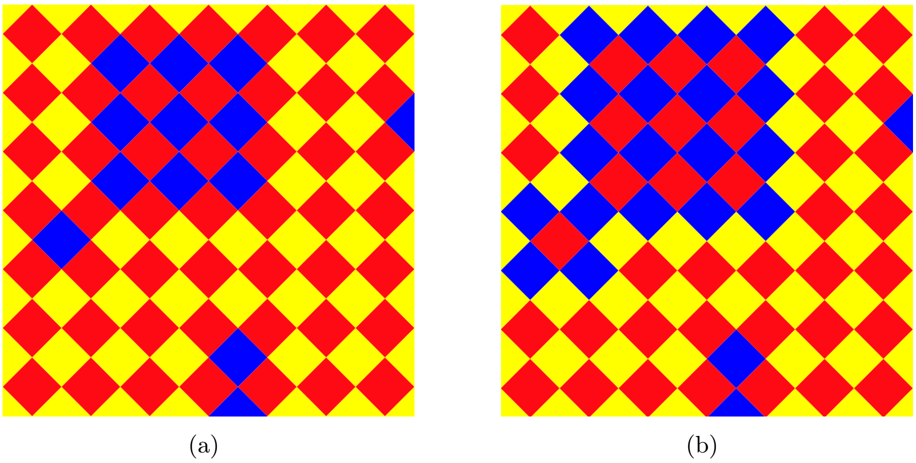


Figure 3.3: *Example of a single cluster Monte Carlo move. A single cluster is constructed from the colours red and blue, after which the colours are interchanged.*

Consider figure 3.3, which illustrates the above algorithm. Step 1 has already been performed, in the way described in section 2.4. Then pick a random plaquette within the cluster. Suppose it is a red one, hence $A =$ 'red'. In this case the second colour was chosen to be blue. The algorithm continues to build the cluster, e.g. the square red and blue area in figure 3.3 with an extension to the bottom left. Every red plaquette in the cluster is then coloured blue, and every blue plaquette is coloured red. The result is figure 3.3(b), which is translated into a new arrow configuration.

Note that as a second colour the algorithm might as well have picked yellow. The original plaquette would then have remained on its own, i.e. in a cluster of a single plaquette. That single plaquette would have been turned yellow. The possibility of these single-plaquette moves is essential to make sure that the algorithm satisfies ergodicity.

Also note that the blue/red cluster is closed off by a boundary of yellow plaquettes. Within this boundary it has a chequerboard pattern. This must be so, since no two neighbouring plaquettes can have the same colour. These two feats are general: the lattice is filled with clusters that are a chequerboard of two colours, surrounded by a boundary of the third colour.

With these remarks, it is time to expand the core of the algorithm to its full version. The basis given above is modified in four ways.

Firstly, the algorithm should be adjusted to account for the energetic behaviour of the F-model. This can be done by adding next-nearest neighbours to the cluster with a probability

$$P_{\text{add}} = 1 - e^{-\beta\epsilon}. \tag{3.11}$$

The motivation behind this probability will be demonstrated in section 3.2.3. Of course, to be added to the cluster the next-nearest neighbour still needs to have colour A or B.

Secondly, the full algorithm covers the entire lattice in clusters of two colours. This is much more effective. After growing a first cluster, the algorithm should attempt to grow another cluster starting from a new plaquette. To this new cluster only those plaquettes can be added that were not considered when growing the first cluster. After this second cluster the algorithm builds a third, etcetera. This continues until all plaquettes with one of the two starting colours have been considered to be part of a cluster.

Thirdly, covering the entire lattice with different clusters can only be done if the colours A and B are picked before building the clusters. Otherwise one could, in the situation where a cluster of colour A and B is bordered by colour C, for example switch $A \rightleftarrows B$ in the cluster, and $B \rightleftarrows C$ in the boundary, only to end up with bordering plaquettes of colour B.

Since this algorithm covers the entire lattice in clusters it cannot interchange the colours in every cluster. For reasons that will become clear in the next section, this would violate ergodicity. Therefore, in the full lattice algorithm the colours in every cluster are only interchanged with a probability of 50%.

With these modifications the full cluster algorithm becomes:

1. Translate the arrow configuration into a three-colour lattice.

2. Pick two different colours A and B.

3. Build a cluster:

   (a) Pick a starting plaquette that has not been considered for any cluster so far.

   (b) Add all neighbouring plaquettes of colour A or B and all next-nearest neighbours of colour A or B with probability P_{add}.

(c) Continue until all possible additions to this cluster have been considered.

4. Build another cluster, until all plaquettes on the lattice have been considered for a cluster.

5. In every cluster, with probability $\frac{1}{2}$ switch colours A and B.

6. Translate the three-colour lattice back to an arrow configuration.

3.2.2 Ergodicity

In order to make sure that the above algorithm produces a sample of configurations that satisfies the Boltzmann distribution, it is necessary to prove ergodicity and detailed balance. Start with ergodicity.

The discussion of ergodicity consists of three parts. First it will be shown that each configuration can, in a finite number of reversible moves, be turned into a configuration in which the entire lattice is covered in a chequerboard of only two colours. The different chequerboard configurations then turn out to be interconnected. These can also, in a finite number of reversible moves, be turned into each other. Together these two statements prove ergodicity of the cluster algorithm on a lattice with open boundary conditions. However, on a periodic lattice ergodicity is more problematic. This will be discussed further below.

Firstly, focus on turning any three-colour configuration into a chequerboard configuration. Any three-colour configuration consists of chequerboard regions surrounded by the third colour. Even a single plaquette of colour C amidst a chequerboard of A and B can be thought of as a very small cluster of C and A plaquettes. That is, if the C-coloured plaquette happens to be surrounded by colour B. Otherwise this plaquette can of course be considered a cluster of colours C and B. For this it is necessary that $P_{\text{add}} < 1$ so that the next-nearest neighbours are not necessarily a part of the cluster of a single plaquette.

This single deviating plaquette in a larger chequerboard region must be surrounded by one and the same colour. Hence, by flipping this very small cluster we can add it to the surrounding A/B chequerboard that is not flipped. For this to be a possible move in the algorithm, it is necessary to not interchange the colours in every cluster. Therefore the colours in each cluster are only interchanged with a probability of 50%. In this way even single blind spots in a chequerboard region can be added to the chequerboard region.

A configuration of one chequerboard region with colours A and B, a boundary of colour C, and another chequerboard region of A and B but with the colours shifted one plaquette, can also be translated into a single chequerboard region. First interchange the colours in the second chequerboard region. Then the border becomes a line of single deviating plaquettes in a large chequerboard region. As we just found, these can also be turned into parts of the chequerboard region.

The shorter way to look at it is to notice that any chequerboard region can be grown to encompass the entire lattice. By clustering and flipping colours, its boundaries can always be moved or added to it. The cluster can always be made larger until it spans the entire lattice. Therefore every possible configuration can be translated into a chequerboard configuration, i.e. a lattice that is completely covered by a chequerboard of only two colours.

Denoting by AB the configuration in which all even numbered plaquettes have colour A and the odd numbered plaquettes colour B, the six possible chequerboard configurations are AB, AC, BA, BC, CA, and CB. Configurations such as AB and BA can of course be turned

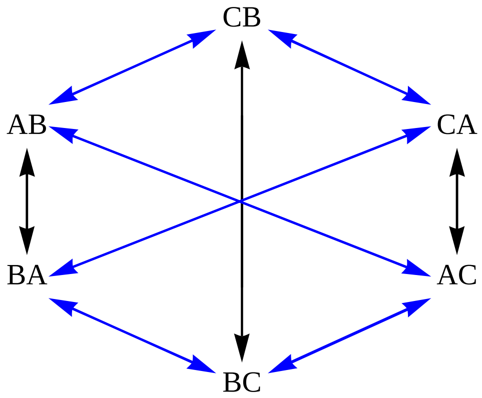


Figure 3.4: *Schematic representation of the transitions between the chequerboard configurations. The black arrows denote transitions by means of a single cluster. In the blue arrows one colour is turned into another by means of a large number of single plaquette clusters, and possibly some clusters involving next-nearest neighbours.*

into each other by the flip of a single cluster that spans the entire lattice. These transitions are denoted by black arrows in figure 3.4. The other transitions completely replace one of the chequerboard colours with the third one. This happens by means of a large number of single cluster operations, or possibly some clusters in which next-nearest neighbours play a part. For example, when turning AB into AC one uses a large number of BC clusters. These processes are denoted by blue arrows. Together, these two mechanisms assure that any chequerboard configuration can be turned into any other chequerboard configuration.

Combining the observations that any configuration can reversibly be turned into a chequerboard configuration and that any chequerboard configuration can reversibly be turned in any other chequerboard configuration, it is proven the clustering algorithm can turn any configuration of the three-colour lattice into any other configuration. However, this proof only works for a lattice with open boundary conditions.

On the other hand, the simulations in this research were conducted on a lattice with periodic boundary conditions. This causes a problem with arrow configurations of which the height modulo three is not periodic, i.e. those arrow configurations that display a height difference between the left and right or top and bottom side of the lattice which is not equal to zero when taken modulo three. When translating between the arrows and the height function modulo three, or three-colour representation, it is unclear how to treat these configurations of which the height modulo three is not periodic. For an example, consider figure 3.5, where the numbers in the plaquettes denote the heights modulo three.

Consider the position of the red 2 in figure 3.5. Assuming periodic boundary conditions, one would expect this position to have height zero (modulo three, that is) because of the 0 on the right-hand side. However, by directly moving to this position from the top left plaquette, where the height is defined to be 1, one obtains the red height of 2. Now should this plaquette be assigned height 2 or 0? Well, if it is assigned height 0, then the arrow right below the red two would be undefined. It would be an arrow between two plaquettes of height 0. Clearly, this cannot be.

However, assigning height 2, and effectively expanding the height lattice by one row is also not an option, since this would allow cluster moves to switch the colours in such a way that the underlying periodicity in the arrow configuration is violated.

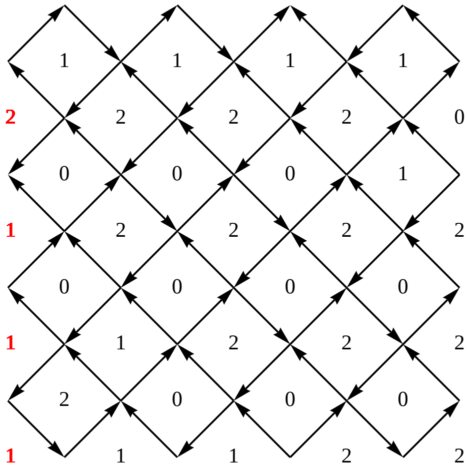


Figure 3.5: *The height function modulo three for a configuration with height difference unequal to zero modulo three. The red heights are obtained by moving to the left from the top left square of height 1, the black heights are obtained by moving to the right.*

Therefore, when working on a periodic lattice, the cluster algorithm cannot deal with these lattice configurations that contain a height difference from side to side that is unequal to zero when taken modulo three. The same holds for similar height differences across the periodic boundary in the vertical direction.

This problem can be solved by making use of both the short loop and the cluster algorithm at the same time. Since the short loop algorithm satisfies ergodicity on the periodic lattice, the combination of the short loop algorithm and the cluster algorithm must also satisfy ergodicity. For if a certain configuration cannot be handled by the cluster algorithm, it can still be handled by the short loop algorithm. Therefore, combining these algorithms will solve this lack of ergodicity.

In practice this means that the Monte Carlo step of the combined algorithm consists of running the short loop algorithm, testing whether there is a height difference unequal to zero modulo three across the sides of the lattice, and then depending on the outcome performing the cluster algorithm, or not performing it. If there is a height difference that prevents the cluster algorithm from being executed, it can be erased by the short loop algorithm in a later Monte Carlo step. After this, the cluster algorithm can also be performed again. However, note that configurations in which the cluster algorithm cannot run are allowed configurations of the F-model, and therefore should still be part of the sampled configurations.

3.2.3 Detailed Balance

For the cluster algorithm to satisfy detailed balance it must obey eq. (3.5). The choice made for P_{add} in eq. (3.11) makes sure that this is the case. This is shown below. Here the detailed balance of the cluster algorithm is treated. Together with the detailed balance of the short loop algorithm this shows that the combined algorithm discussed in the last section also satisfies detailed balance.

Consider $P(\mu \to \nu)$, the probability to move from configuration μ to configuration ν under one Monte Carlo step of the clustering algorithm. It consists of the probability to choose two specific colours to interchange, $\frac{1}{3}$, the probability of deciding to flip these specific clusters out of the n clusters the lattice is covered with, 2^{-n}, a product over the different clusters, and per cluster a factor P_{add} for every plaquette that is only a next-nearest neighbour and has been added to the cluster, and a factor $1 - P_{\text{add}}$ for every next-nearest neighbour that did not make it to the cluster. In short

$$P(\mu \to \nu) = \frac{2^{-n}}{3} \prod_{\substack{\text{Flipped}\\ \text{clusters}}} \prod_{(i,j)_{con}} P_{\text{add}} \prod_{(i,j)_{discon}} (1 - P_{\text{add}})^{\delta\left(c_i^\mu, c_j^\mu\right)}. \tag{3.12}$$

In this $\delta\left(c_i^\mu, c_j^\mu\right)$ is the Kronecker delta function which has value 1 or 0, depending on whether or not the colour of plaquette i in configuration μ (c_i^μ) is the same as that of plaquette j in configuration μ (c_j^μ). Moreover $(i,j)_{con}$ signifies that the product is to be taken over the pairs of next-nearest plaquettes that are connected in the cluster, whilst the product that runs over $(i,j)_{discon}$ covers all next-nearest disconnected pairs, those pairs of which only one made it to the cluster.

Taking the logarithm puts eq. (3.12) in a more usable form:

$$\begin{aligned}\log\left[P(\mu \to \nu)\right] = \sum_{\substack{\text{Flipped}\\ \text{clusters}}} \Bigg(\log\left[\frac{2^{-n}}{3}\right] + \log\left[P_{\text{add}} \sum_{(i,j)_{con}} 1\right] \\ + \log\left[1 - P_{\text{add}}\right] \sum_{(i,j)_{discon}} \delta\left(c_i^\mu, c_j^\mu\right) \Bigg).\end{aligned} \tag{3.13}$$

The aim is to verify eq. (3.5). Its left-hand side can be filled in after taking the logarithm. Since almost all terms in the above eq. (3.13) are the same for $P(\nu \to \mu)$ many terms cancel. One finds

$$\begin{aligned}\log\left[\frac{P(\mu \to \nu)}{P(\nu \to \mu)}\right] &= \log\left[P(\mu \to \nu)\right] - \log\left[P(\nu \to \mu)\right] \\ &= \log\left[1 - P_{\text{add}}\right] \sum_{(i,j)_{discon}} \delta\left(c_i^\mu, c_j^\mu\right) - \delta\left(c_i^\nu, c_j^\nu\right).\end{aligned} \tag{3.14}$$

Now take a look at the right-hand side of the detailed balance equation (3.5), and consider the expression for $\Delta E = E_\nu - E_\mu$. The energy difference between the two configurations is determined by the change in the number of symmetrical vertices. In the three-colour representation a symmetric vertex looks like a tiny chequerboard, i.e. it consists of two next-nearest neighbour pairs of same coloured plaquettes. The only contributions to this energy difference can come from the edge of a cluster, since in the heart of a cluster all symmetric vertex states will be symmetric in both configuration μ and ν. Hence the energy difference is determined by the number of same coloured next-nearest neighbour pairs of which one belongs to one cluster, and the other one to another cluster. Every new symmetric vertex contributes $-\epsilon$, so that the energy difference becomes

$$\begin{aligned}\Delta E &= E_\nu - E_\mu \\ &= -\epsilon \sum_{\substack{\text{Flipped}\\ \text{clusters}}} \sum_{(i,j)_{discon}} \delta\left(c_i^\nu, c_j^\nu\right) - \delta\left(c_i^\mu, c_j^\mu\right) \\ &= \epsilon \sum_{\substack{\text{Flipped}\\ \text{clusters}}} \sum_{(i,j)_{discon}} \delta\left(c_i^\mu, c_j^\mu\right) - \delta\left(c_i^\nu, c_j^\nu\right).\end{aligned} \tag{3.15}$$

Substituting this in eq. (3.14) gives

$$\log\left[\frac{P(\mu\to\nu)}{P(\nu\to\mu)}\right] = \log\left[1-P_{\mathrm{add}}\right]\frac{\Delta E}{\epsilon}. \tag{3.16}$$

The detailed balance equation (3.5) is satisfied as long as

$$\log\left[\frac{P(\mu\to\nu)}{P(\nu\to\mu)}\right] = -\beta\Delta E. \tag{3.17}$$

Hence detailed balance is satisfied if

$$-\beta\Delta E = \log\left[1-P_{\mathrm{add}}\right]\frac{\Delta E}{\epsilon}. \tag{3.18}$$

That is, detailed balance is satisfied as long as

$$P_{\mathrm{add}} = 1 - e^{-\beta\epsilon}. \tag{3.19}$$

Compare eq. (3.11).

Since both the short loop algorithm and the cluster algorithm satisfy detailed balance, so will the combination described at the end of section 3.2.2. This combined algorithm was used to run the simulations reported on in the following chapters.

Chapter 4

Finite-Size Scaling

Outline

This chapter treats the technique of finite-size scaling and its application to the F-model. Firstly, the method of finite-size scaling is discussed theoretically, using the two-dimensional Ising model as an example. After this, finite-size scaling is applied to the F-model, obtaining a remarkable result for the critical temperature, which is then discussed.

4.1 Finite-Size Scaling in the Ising Model

Simulations of phase transitions in which a correlation length diverges have an inevitable disadvantage, namely finite-size effects. Simulations are run on finite lattices, and this will affect the outcome. Especially when modelling phase transitions, for which the theoretical predictions usually rest on taking the limit towards an infinite system, the relatively small size of the simulated model will affect the obtained data. These effects are reduced by applying periodic boundary conditions, but of course only to a limited extent.

The power of finite-size scaling is that it uses these finite-size effects to make predictions for infinite systems. By using some of the relations known from the theoretical approach of the model, computational data can be combined in such a way as to give quite exact measurements for the coefficients, free from finite-size effects.

This section illustrates the method of finite-size scaling by applying it to the two-dimensional ferromagnetic Ising model, much in the way done by Barkema and Newman [35]. In the next section this is applied to the F-model.

To see how finite-size scaling works, recall the two-dimensional ferromagnetic Ising model, and consider the magnetic susceptibility. Using the reduced temperature t

$$t \equiv \frac{T - T_\mathrm{c}}{T_\mathrm{c}} = \frac{\beta_\mathrm{c}}{\beta} - 1, \tag{4.1}$$

the magnetic susceptibility χ diverges algebraically near the critical temperature as

$$\chi \sim |t|^{-\gamma}, \tag{4.2}$$

where γ is a critical coefficient of the Ising model.

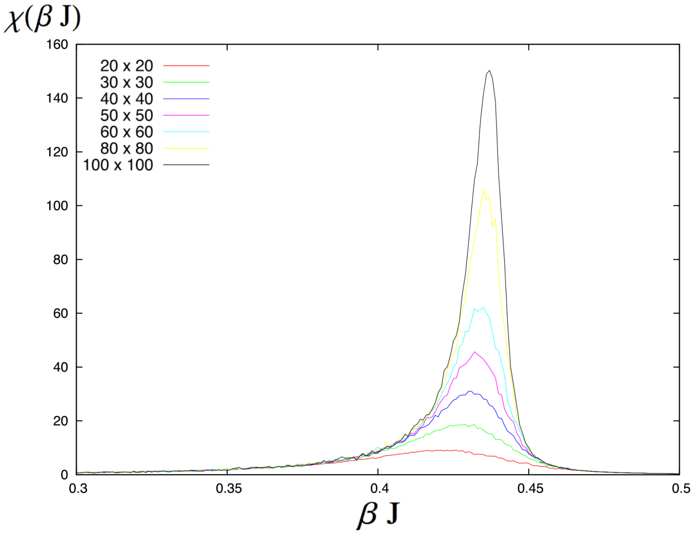


Figure 4.1: *Divergence of the susceptibility of the ferromagnetic two-dimensional Ising model for various lattice sizes. Notice how the divergence better resembles a true divergence to infinity for larger lattice sizes.*

This divergence is plotted for several simulations with different lattice sizes in figure 4.1. The finite-size effects are clearly visible. The divergence of the susceptibility resembles a true divergence to infinity better as the size of the simulated lattice becomes larger. This is the finite-size behaviour that needs to be overcome, or even better, used to our advantage.

Now observe the correlation length ξ, that near criticality diverges as

$$\xi \sim |t|^{-\nu}. \tag{4.3}$$

From eqs. (4.2) and (4.3) the divergence of the susceptibility can be expressed in terms of the correlation length:

$$\chi \sim |\xi|^{\gamma/\nu}. \tag{4.4}$$

However, in a system of finite size there can be no regions of correlating spins that are larger than the system itself, so that the correlation length is cut off, $\xi < L$. As a consequence, the susceptibility is also cut off, and instead of a real divergence towards infinity the simulation produces results such as in figure 4.1.

Now account for this cut-off by writing

$$\chi \sim \xi^{\gamma/\nu} \chi_0 \left(\frac{L}{\xi} \right). \tag{4.5}$$

In the above expression ξ is still the correlation length as it would behave in an infinite system, but the cut off takes place as $\xi > L$. Therefore for $\xi \ll L$ the simulated system and the infinite system should display the same χ.

This behaviour is accounted for by taking

$$\chi_0(x) \begin{cases} = \text{constant}, & \text{if } x \gg 1 \\ \sim x^{\gamma/\nu}, & \quad \text{as } x \to 0. \end{cases} \tag{4.6}$$

This function incorporates the essential scaling behaviour that is needed. However, it is still hard to measure, since it still contains ξ, the correlation length in an infinite system.

By introducing the dimensionless variable

$$\widetilde{\chi}(x) \equiv x^{-\gamma}\chi_0(x^{\nu}), \tag{4.7}$$

eq. (4.5) can be rewritten into

$$\chi \sim \xi^{\gamma/\nu}\chi_0\left(\frac{L}{\xi}\right) = \xi^{\gamma/\nu}\left(\frac{L}{\xi}\right)^{\gamma/\nu}\tilde{\chi}\left(L^{\frac{1}{\nu}}\right) = L^{\gamma/\nu}\widetilde{\chi}\left(L^{1/\nu}t\right). \tag{4.8}$$

The beauty is that in this equation all dependencies on system size are explicitly expressed. This will be very helpful in a moment.

But first note that so far the asymmetrical behaviour on both sides of the criticality has not been not accounted for. It can be incorporated by extending the definition of $\widetilde{\chi}(x)$ to negative x. A slight further rearrangement gives

$$\widetilde{\chi}\left(L^{1/\nu}t\right) = L^{-\gamma/\nu}\chi(t). \tag{4.9}$$

This is the scaling function to be measured. Since all dependencies on system size are explicitly expressed eq. (4.9) should equally hold for all simulations, provided each simulation is processed using its specific L. That is, the curves only collapse for the correct values of γ, ν, and T_{c}, or β_{c}. Making sure that the curves for various system sizes collapse to a single curve is therefore a way of determining γ, ν, and T_{c}, or β_{c}. In this case the best collapse takes place when $\gamma = 1.75$, $\nu = 1.00$ and $T_{\mathrm{c}} = 2.268J(\beta_{\mathrm{c}}J = 0.441)$. The result is shown in figure 4.2.

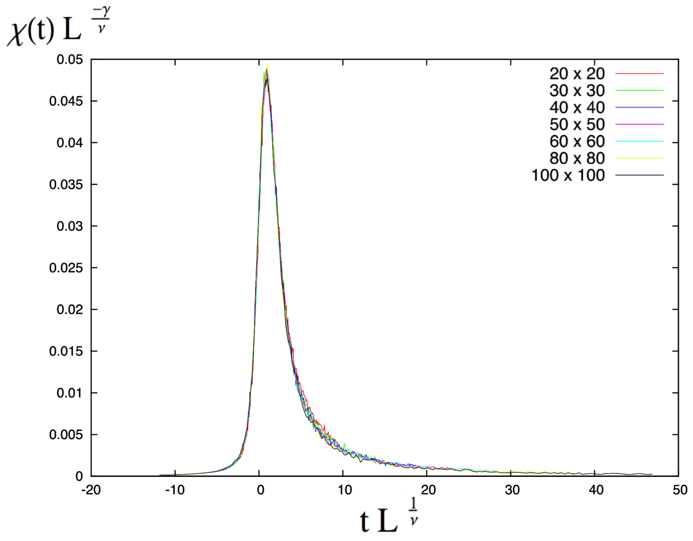


Figure 4.2: *Finite-size scaling for the two-dimensional Ising model. Using the right coefficients the curves of eq.* (4.9) *collapse. This plot is made using* $\gamma = 1.75$, $\nu = 1.00$ *and* $T_c = 2.268J(\beta_c J = 0.441)$.

4.2 Finite-Size Scaling in the F-model

After this example of finite-size scaling in the Ising model, the question comes up how to apply it in the F-model. In fact, since for now the only goal is to obtain the critical temperature, this scaling will be simpler than the one performed last section.

The quantity under consideration is $P_{\text{perc}}(\epsilon\beta)$, i.e. the probability of, at a given inverse temperature $\epsilon\beta$, finding a ground state region that wraps around the periodic boundaries and connects to itself, i.e. percolates. Recall figure 2.5, on page 14. This shows that for low temperatures, or high $\epsilon\beta$, there is one large ground state region covering the entire lattice. It will definitely percolate. As the temperature rises the ground state regions become smaller, and the probability for one of them to percolate should become less and less.

Now consider figure 4.3 that displays the measured $P_{\text{perc}}(\epsilon\beta)$ for various lattice sizes.

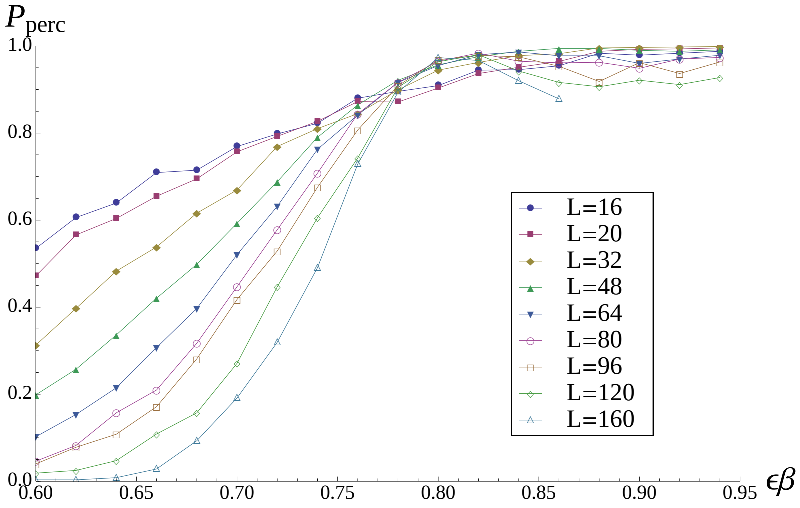


Figure 4.3: *The measured $P_{perc}(\epsilon\beta)$ for various lattice sizes. The behaviour is as expected when compared to figure 2.5.*

In the above graph the finite-size effects are clear. The larger the simulated system becomes, the closer it gets to the theoretical picture which is that of a step function.

The goal is to apply finite-size scaling. Because the finite-size effects are closely connected to the divergence of the correlation length, first recall its divergence from eq. (2.14). For now it is more instructive to write, for a some constant,

$$\xi(\epsilon\beta) \sim e^{a\sqrt{\epsilon\beta}}. \tag{4.10}$$

This is equivalent to

$$\frac{\log\left[\xi(\epsilon\beta)\right]^2}{\epsilon\beta} \sim a^2. \tag{4.11}$$

The above equation shows that the finite-size behaviour enters with the square of the logarithm of the length scale. This behaviour should be corrected for by multiplying the (inverse) temperature with the same factor, i.e. $(\log[L])^2$. Furthermore, the inverse temperature is rescaled by looking at its deviation from the critical value. Together, the finite-size scaling can be summarized by

$$\epsilon\beta \Longrightarrow \log\left[L\right]^2 \frac{\epsilon\beta - \epsilon\beta_{\text{c}}}{\epsilon\beta_{\text{c}}}. \tag{4.12}$$

As a function of this new rescaled temperature the graphs of P_{perc} for various lattice sizes are expected to collapse. That is, as long as the right $\epsilon\beta_{\text{c}}$ is used. This amounts to a measurement of $\epsilon\beta_{\text{c}}$.

The best collapse of our data takes place for $\epsilon\beta_{\text{c}} = 0.79 \pm 0.02$, as is shown in figure 4.4. The estimation of accuracy, 0.02, was obtained by varying $\epsilon\beta_{\text{c}}$ and considering for which values the collapse was distinguishably better.

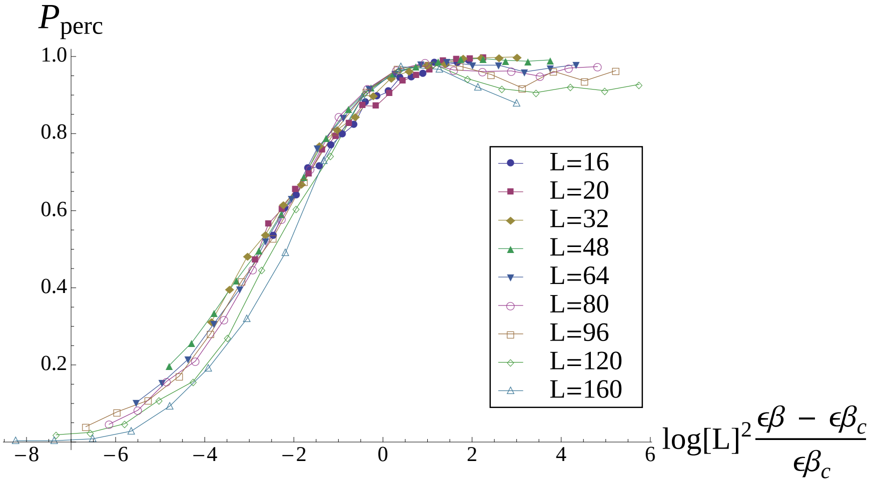


Figure 4.4: *The rescaled* $P_{perc}(\log[L]^2 \frac{\epsilon\beta - \epsilon\beta_c}{\epsilon\beta_c})$ *for various lattice sizes. The curves collapse for* $\epsilon\beta_c = 0.79 \pm 0.02$.

4.3 Discussion

The collapse of figure 4.4 was produced using $\epsilon\beta_{\text{c}} = 0.79 \pm 0.02$. The difference from the analytical value $\epsilon\beta_{\text{c}} = \log[2] \approx 0.693$ calls for an explanation. Let us consider four possibilities.

Firstly, it is of course possible that the analytical value is not correct. However, this does not seem very likely. Lieb's solution [4] has widely been accepted as the correct approach to this model. Furthermore, it has successfully been elaborated on by Baxter [8], whose work also remains unchallenged. It seems much more likely that the difference in values for $\epsilon\beta_{\text{c}}$ can be explained by one of the mechanisms below, than that the work of Lieb, Baxter and many fellow theorists after them needs to be rewritten.

Secondly, it was verified that Lieb and Baxter found their solution on the same kind of square lattice, also with periodic boundary conditions in both directions. This cannot account for the difference in $\epsilon\beta_{\text{c}}$.

Thirdly, it is of course possible that the algorithms used either are inherently incorrect, or at the very least were incorrectly implemented. The first suggestion seems unlikely. For the algorithms to be incorrect, i.e. to sample the configurations incorrectly, one of the proofs in the previous chapter should be wrong. This is of course possible, but it seems unlikely. Except for the part on configurations with a height difference unequal to zero modulo three, they were developed and published long before this research.

An incorrect implementation of the presented algorithms is an option that has to be considered seriously. The full program used contains over 3800 lines of code. It is a possibility that one or more of these contain an error that affects the simulation in such a way as to produce the wrong critical temperature. However, this possibility was tested for by separately running the cluster algorithm and the loop algorithm. That is, as much as was possible, because as shown in section 3.2.2 the cluster algorithm does not function without the short loop algorithm to take it out of certain problematic configurations. These simulations of both algorithms separately did not show any relevantly different behaviour from the combined simulation, or from each other for that matter. As it seems unlikely that in the implementation of both algorithms errors were made with the same effect, the difference in critical temperatures is not expected to be caused by an error in implementation.

Another possibility is that the finite-size scaling was not performed correctly. The argument based on eq. (4.11) seems strong, but it can also be looked at from another direction. One could also consider the behaviour of $\frac{\xi}{L}$ as the crucial ingredient for scaling. However, considering this fraction does not produce the scaling behaviour the above analysis counts on. Trying to bring the division by L into the inverse temperature results in

$$\frac{\xi\left(\epsilon\beta\right)}{L} \sim \frac{e^{a\sqrt{\epsilon\beta}}}{L} = e^{\log\left[1/L\right]}e^{a\sqrt{\epsilon\beta}} = e^{a\sqrt{\epsilon\beta}-\log\left[L\right]}. \tag{4.13}$$

This suggests that the finite-size effects cannot be corrected for by simply multiplying with some factor that depends on L. Therefore the scaling performed above may be thought of as too simple.

However, a strong argument against this is the fact that the curves do collapse. If eq. (4.13) means that the F-model cannot be treated by finite-size scaling, or at least not in the way of eq. (4.12), then it seems strange that the curves in figure 4.4 clearly collapse. This can hardly be blamed on coincidence. One would need to provide a new explanation for this if the finite-size scaling cannot be performed as done in eq. 4.12.

Moreover, the measurements of the probability of two vertices to be in the same ground state region presented in the next chapter (figure 5.8) also suggest a critical temperature around $\epsilon\beta = 0.8$. On these measurements no finite-size scaling was applied.

In conclusion, it is hard to give a definite answer to the question where the difference between the analytical value and the result of the performed finite-size scaling stems from. Several suggestions can be made, but most of them do not seem to be very strongly founded. The question remains whether the finite-size scaling should be based on eq. (4.11) or eq. (4.13). If the answer is the latter, then the fruitful application of the rescaling (4.12) needs a new explanation.

Chapter 5

Real Space Renormalization

Outline

The aim of this chapter is to obtain critical coefficients of the F-model by applying real space renormalization. Firstly, the framework of real space renormalization is set out and illustrated by its application to the Ising model. Then the focus is moved towards the F-model, and the complications that arise in the coarse graining of it. These complications can be overcome, and a coarse graining prescription is presented. Its application is followed by a discussion of its effectiveness.

5.1 Theory

5.1.1 Basic Notion

The basic notion of real space renormalization is that one can consider a system with a discrete scaling symmetry and reduce the degrees of freedom in order to find a smaller state which can again be considered a state of the same system, but now at a different length scale. The new state is a state of the same type of system, but some parameters have changed. One of these is the temperature. The renormalized state, as the state after this procedure is called, turns out to represent the considered system at a slightly different temperature, or with for instance different coupling constants. From the way these parameters change under renormalization the critical behaviour of the system can be determined.

In the core, this is what real space renormalization means. In this chapter we only consider parameter flow in one direction, that of the temperature. This is sufficient for the study of the F-model.

The basic thought of real space renormalization was introduced by Kadanoff [39] in 1966, but it took the work of Wilson, Kogut and Fisher [40, 41, 42] in the early seventies to demonstrate the full potential of this technique and to expand it to the renormalization group. This idea rests on the repeated application of the procedure described above. Since the state after renormalization can be treated as a state of the original system, one can again perform the same procedure. Repeating the renormalization over and over results in a series of states with a corresponding flow in the parameter space. This is mathematically a semi-group, and by physicists referred to as the renormalization group.

At this point it is important to stress that real space renormalization is different from the techniques of renormalization used in field theory. These techniques are related, yet quite different in the sense that the degrees of freedom which are integrated over usually lie in momentum space. Renormalization in this thesis refers to real space renormalization.

The approach taken in this research is that of the Monte Carlo renormalization group, as first developed by Ma [43] and Swendsen [44]. The Monte Carlo renormalization group is based on real space renormalization, but fills in the renormalization framework by performing simulational measurements rather than a fully analytical approach.

There are many excellent texts available on (real space) renormalization and the renormalization group. The treatment below is based on the accounts of Binney et al. [45], Cardy [46], Goldenfeld [47], McComb [48], and Newman & Barkema [35]. For a more detailed account, the reader is referred to these texts.

5.1.2 Coarse Graining

In order to perform any renormalization one first needs a procedure of coarse graining the states of the system of interest. That is, we are looking for a procedure to generate a renormalized state, one with fewer degrees of freedom, starting from any original state. An example of such a procedure for the two-dimensional Ising model is illustrated in figure 5.1.

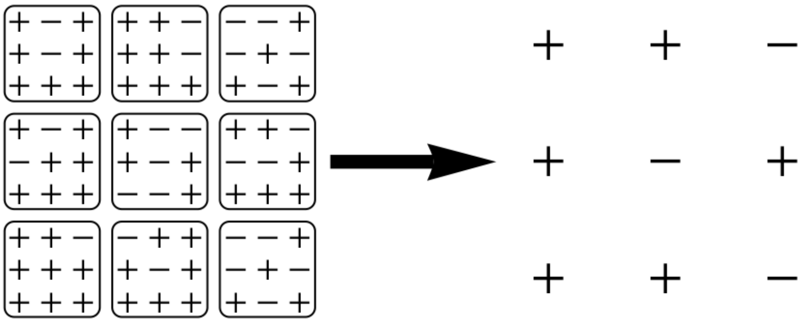


Figure 5.1: *A coarse graining procedure by majority rule applied to the two-dimensional Ising model on a square lattice. The regions of* 3×3 *spins are replaced by a single block spin each.*

The spins in the original state are first grouped in regions of nine spins. These nine spins are then replaced by a single block variable, in this case a block spin, the value of which is determined by the majority of the spins in the corresponding block of the original lattice. This so called majority rule is a commonly used way to determine the value of the block spin from the original spins.

The majority rule is not the only option to base a coarse graining prescription on. Other coarse graining procedures often employed are those of decimation, where the value of, for example, the top left spin in the corresponding block is taken as the block spin; and averaging, where average value of all variables in the block is taken.

The coarse graining procedure should reproduce the overall characteristics of the state that is renormalized. This behaviour is visible in figure 5.1. The background of mostly $+$-states is reproduced, and also the band of mostly $-$-states that flows from the top right through the middle block to the bottom right is visible in the coarse grained state.

The size of the lattice is reduced by a factor of three in this coarse graining. This factor is called the rescaling factor. It is denoted by b. The rescaling factor can take other values, but it usually lies in this order of magnitude.

5.1.3 The Correlation Length Under Renormalization

To gain some insight into the behaviour of the system under renormalization it is instructive to consider the correlation length. The Ising model again serves as a reference. Its behaviour under renormalization is qualitatively represented in figure 5.2.

Under renormalization all length scales in the system are reduced by a factor b, so that the same should happen to the correlation length:

$$\xi \quad \Longrightarrow \quad \xi' = \frac{\xi}{b}. \tag{5.1}$$

As illustrated in figure 5.2, the change in correlation length under renormalization is associated to a change in temperature. This is the movement in parameter space discussed earlier. The renormalized system can again be considered a state of the same system, but at different temperature; as ξ becomes ξ', also T becomes T'.

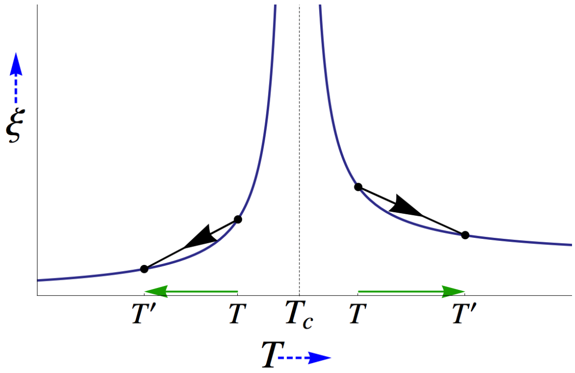


Figure 5.2: *Qualitative behaviour of the correlation length of the Ising model under renormalization. As the correlation length is reduced the temperature flows away from the critical point.*

Moreover, it is important to note that the temperature flows away from the critical value. For temperatures below T_c the temperature of the renormalized system is lower. Below the critical temperature the spins in the Ising model will on average have some spontaneous magnetization. As the renormalization is applied repeatedly, and larger and larger regions are replaced by block spins, the complete system will be assigned this direction of magnetization. Effectively, the repeated renormalization strengthens a small preference for a spin direction until it covers the entire lattice. Such a system with all spins pointing in one direction is typical of very low temperatures in the ferromagnetic Ising model. Therefore, under repeated renormalization any temperature $T < T_c$ flows to even lower temperatures; $T' < T$.

For temperatures above T_c the renormalized temperature is even higher. Since the correlation length becomes smaller under renormalization, the block spins become less and less correlated. Under repeated renormalization the correlation length goes to zero, and the spins are randomly distributed. This means that for temperatures $T > T_c$ the renormalized temperature will be even higher: $T' > T$.

The ‘watershed’ is T_c, the position relative to this determines whether under renormalization the temperature will increase or decrease. See figure 5.3.

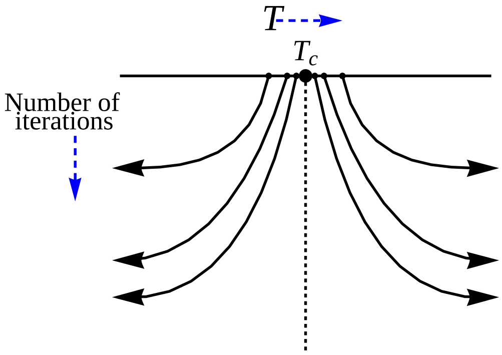


Figure 5.3: *Flow of the temperature under repeated renormalization. The flow towards high and low temperature is separated by the ‘watershed’ at T_c. Picture adapted from Barkema [49].*

If the correlation length is either zero or infinite, dividing it by a factor b will result in a renormalized correlation length that is the same as the original one. Therefore, if the temperature is either 0, T_c, or ∞, the temperature after renormalization is the same. These points are called fixed points.

The fixed points of the renormalization at $T = 0$ and $T = \infty$ can physically be explained in the following way. At infinite temperature, the spins in the Ising lattice are randomly oriented. Any sensible coarse graining procedure will translate this into randomly oriented block spins. This means that the renormalized temperature is again infinite. Hence, if the system is at infinite temperature, it will stay there under renormalization. At the other end, when $T = 0$, the system is completely ordered. The coarse graining procedure will reproduce a completely ordered system, which has the same temperature. Again, this is a fixed point.

The watershed point at $T = T_\mathrm{c}$ in fact is a critical surface in the parameter space. A state on this critical surface will generally be sent to a different set of parameters, but also on the critical surface. Note that for this different set of parameters, in the Ising model for instance a different spin coupling, the critical temperature will also change. This research is performed in a one-dimensional parameter space, only including the temperature, or to be more precise the variable $\epsilon\beta$. The value of this fully determines the behaviour of the F-model.

5.1.4 Extracting Coefficients

The critical coefficients of the system can be determined from the flow of the parameters under renormalization. Again consider the correlation length of the Ising model. Under renormalization this should behave as

$$\xi \sim |t|^{-\nu} \quad \Longrightarrow \quad \frac{\xi}{b} = \xi' \sim |t'|^{-\nu}, \tag{5.2}$$

where t is the reduced temperature as defined in eq. (4.1). Note that the coefficient ν does not change under renormalization. Its constant value is exactly what is typical about the way the Ising model, and all other systems in the same equivalence class, behave under renormalization.

Equation (5.2) means that

$$\left(\frac{t}{t'}\right)^{-\nu} = b. \tag{5.3}$$

Since b and t are known, the above equation shows that ν can be determined by measuring t'. This is the route we intend to follow.

The description of ξ used in eq. (5.2) is only valid near the critical temperature. Hence, the measurements should focus on this area. In this region the temperature change can be linearised by

$$T' - T_{\mathrm{c}} = (T - T_{\mathrm{c}}) \left.\frac{dT'}{dT}\right|_{T_{\mathrm{c}}}. \tag{5.4}$$

Substituting this into eq. (5.3) gives the final expression by which to find ν,

$$\nu = \frac{\log[b]}{\log\left[\frac{dT'}{dT}\right]\big|_{T_{\mathrm{c}}}}. \tag{5.5}$$

Hence, it is necessary to determine T', the temperature after renormalization. In our simulational approach this is done by measuring some monotonic function $f(T)$ of the temperature before and after renormalization. Denoting the value of f in the renormalized state that before renormalization had a temperature T by $f'(T)$, the renormalized temperature obeys

$$f'(T) = f(T'). \tag{5.6}$$

Since f is a monotonic function it can be inverted. The temperature after renormalization is then determined by

$$T' = f^{-1}\left(f'(T)\right). \tag{5.7}$$

This summarizes the basic simulational approach of real space renormalization in the Ising model. For a more detailed analysis see the work of Swendsen [50], Blöte [51], and the literature cited in section 5.1.1. This research focuses on the question how to apply the framework of real space renormalization to the F-model.

5.2 Real Space Renormalization of the F-model

5.2.1 Coarse Graining Problem

In order to apply real space renormalization to the F-model it is necessary to develop a method of coarse graining. The coarse graining method should be able to reduce any configuration of the F-model to a corresponding configuration of the F-model on a smaller lattice. Also, the new configuration should reproduce the general features of the original configuration.

At first one might consider a block-variable approach similar to that of figure 5.1. However, this is not possible for the F-model, since the states of neighbouring vertices are not independent. In the Ising model, one can renormalise block by block because any two block spins can be neighbours. If one block is replaced by a + while its neighbour is replaced by a − this does not mean the state is impossible. At most it will create an energetic punishment.

This is not the case in the F-model. Replacing different blocks of vertices by one vertex each, one will inevitably place vertex states that cannot be neighbours next to each other. See figure 5.4, where an attempt was made to renormalise the black arrow configuration by renormalized blocks of four vertices that were then replaced by the vertex state (green arrows) that appears most often in each block. This leads to a conflict due to incompatible neighbouring vertex states. The red arrow has to point both ways, thus violating the ice rules. Such 'local renormalization' is not possible in the F-model.

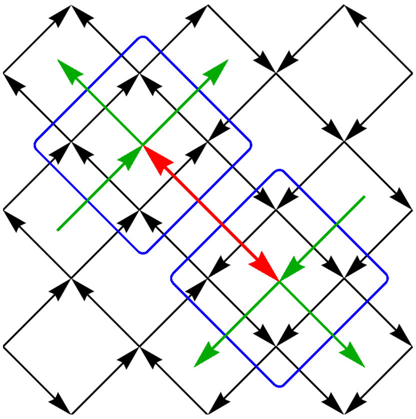


Figure 5.4: *The impossibility of coarse graining the F-model by block variables. In both blue blocks the four vertices are replaced by the vertex state occurring most often, which forces the red arrow to point in both directions. This violates the ice rules.*

5.2.2 Coarse Graining Prescription

The impossibility of a coarse graining procedure by block variables does not mean it is impossible to coarse grain the F-model. In this research we did find a coarse graining prescription that reduces any configuration of the F-model to a configuration on a smaller lattice that is again an allowed configuration of the F-model, and displays the overall features of the original configuration. It is based on the height function described in section 2.3.

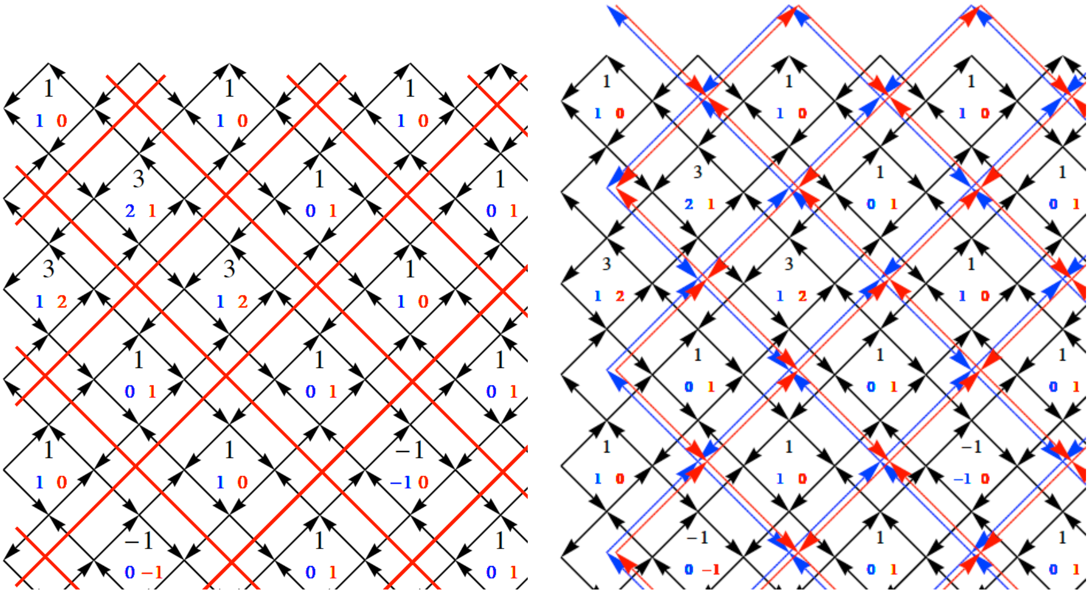
The coarse graining prescription consists of five steps. For an illustration see figure 5.5.

The first step is to determine the bonds of the renormalized lattice, and the heights of the now isolated central plaquettes. The bonds of the renormalized lattice are shown as red lines in figure 5.5(a). They divide the original lattice in cells of which we focus on the central plaquettes. Since between the central plaquettes there are two (anti)-parallel arrows on the original lattice these central plaquettes must always have a height difference of 0, 2 or -2. Therefore, picking a height of 1 for the top left plaquette will ensure that all the central plaquettes have an odd height. This is needed for the further procedure.

Another possibility is to place the renormalized lattice one lattice spacing to the right. All central plaquettes, including the top left plaquette that is assigned height 1, then shift one lattice spacing to the right. This will result in different renormalized configurations. Therefore, the renormalization was in fact performed for both possible positions of the renormalized bonds. Each positioning creates two renormalized lattices, as demonstrated below. The thus obtained four renormalized lattices were averaged over in the measurements. As they are completely equivalent only the renormalization as shown in figure 5.5(a) is discussed here.

The next step is to divide the heights of the central plaquettes by two. Since they are all odd, this produces half-integers. Write down both the integer value when rounded off upwards, and when rounded off downwards. These are the red and blue numbers in figure 5.5.

From the red and blue numbers the two renormalized configurations of the F-model can be constructed. Starting, for example, at the red 0 in the top left corner, the red lattice heights



(a) (b)

Figure 5.5: *The coarse graining procedure of the F-model. On the left the red lines signify the bonds of the renormalized lattice, the black numbers are the original height in each plaquette, the blue and red numbers signify the heights of the two renormalized lattices. From these the renormalized arrow configuration shown on the right can be constructed.*

are determined by walking over the lattice, always picking from the divided and rounded off heights the one height that is exactly 1 higher or lower than the previous height. In figure 5.5 the heights of these two new lattices are coloured red and blue respectively.

If one starts at the red 0 in the top left corner, then the height of the bottom right neighbour should be 1, and not 2. Coming from this plaquette of (red) height 1, the plaquette one further down and to the right should have height 2, since it must be 1 different from the height 1. In this way the full lattice of red heights can be determined. The other heights, the blue ones, also form a complete lattice with a height difference of 1 between neighbouring plaquettes.

From these blue and red height lattices one can again construct arrow configurations by reversing the height function prescription described in section 2.3. That is, moving from one plaquette to the other, the arrow crossed points to the right if the new height is higher than the previous one, and to the left if the new height is lower. The renormalized arrow configurations are shown in figure 5.5(b).

Note the connection between the renormalized arrows and the original black arrows. When moving from one central plaquette to another, the two black arrows crossed in between can either point parallel or anti-parallel. If they point parallel, then the height difference on the original lattice is 2, both the red and the blue lattice make the same step in height, and both renormalized arrows point in the same direction as the arrows before renormalization. The other option is that the unrenormalized arrows are anti-parallel. The height difference between the central plaquettes then is 0. In this case the red and blue lattice switch relative places. That is, the higher one of the two becomes the lower one, and vice versa. One arrow in the renormalized configuration takes the direction of the arrow first crossed in the original lattice, the other renormalized arrow takes the value of the second one. At the next such transition, where the unrenormalized arrows again point anti-parallel, the relative heights are interchanged and the renormalized lattice that before got its arrow from the firstly crossed black arrow now shares it with the secondly crossed and vice versa.

This change in relative height between the red and blue renormalized lattices can create a problem with the periodic boundary conditions. There are arrow configurations such that in a specific central plaquette at first the red lattice is higher, but after a walk over the periodic boundaries, coming back to the same plaquette, the blue lattice is the relatively higher one. This means that stepping from such a position to a neighbouring central plaquette across two anti-parallel black arrows, the orientation of the red and blue arrows depends on the number of times one has crossed the periodic boundaries. This is not desirable. It can be solved by first doubling the original lattice before renormalizing it. That is, take four copies of the original lattice, so that the lattice length L is doubled, and then renormalise this doubled lattice. In this way the renormalized lattice will again satisfy the same periodic boundary conditions as the original lattice did.

The full renormalization procedure can be summarized by:

1. Double the considered configuration by joining four copies of the configuration together.
2. Draw the bonds of the renormalized lattice, and determine the heights in the central plaquettes, such that they are all odd.
3. Divide all heights by 2, and round off to both nearest integers.
4. Determine the heights of the renormalized lattices by starting from one of the new heights, and walking around the lattice setting the heights such that there is always a height difference of 1 between neighbouring plaquettes.
5. Determine the renormalized arrow configuration based on the heights of the two new lattices.

As noted earlier, using this procedure one can renormalise any configuration of the F-model into a smaller configuration of the F-model, that again satisfies periodic boundary conditions. The question now is to what extent it reproduces the features of the original configuration. As the basis of the coarse graining is to simply divide all heights by 2, the original configuration should be clearly recognizable in the renormalized ones. However, not each height difference is reproduced in the renormalized configurations since the division by two and rounding off means that the renormalized heights can skip over a height difference of 1 between two plaquettes.

An example of the effect of the coarse graining procedure is shown in figure 5.6. The general features of the original lattice are reproduced in the renormalized configurations.

For another illustration of the functioning of the coarse graining, consider the smallest possible excitation from the ground state. This configuration looks like figure 2.4 on page 13, but with one loop of four arrows flipped, as in figure 3.1 on page 20. If the loop of flipped arrows coincides with a central plaquette, one of the two renormalized lattices will not be in the ground state. Four of its vertices, those around the loop, become asymmetric. This is the blue lattice in figure 5.7(a). The red one is not changed by this defect and still shows a ground state. Also, the other two renormalized lattices are not changed by this excitation.

If the defect is not such a central tile, but crosses the positions of the renormalized bonds,

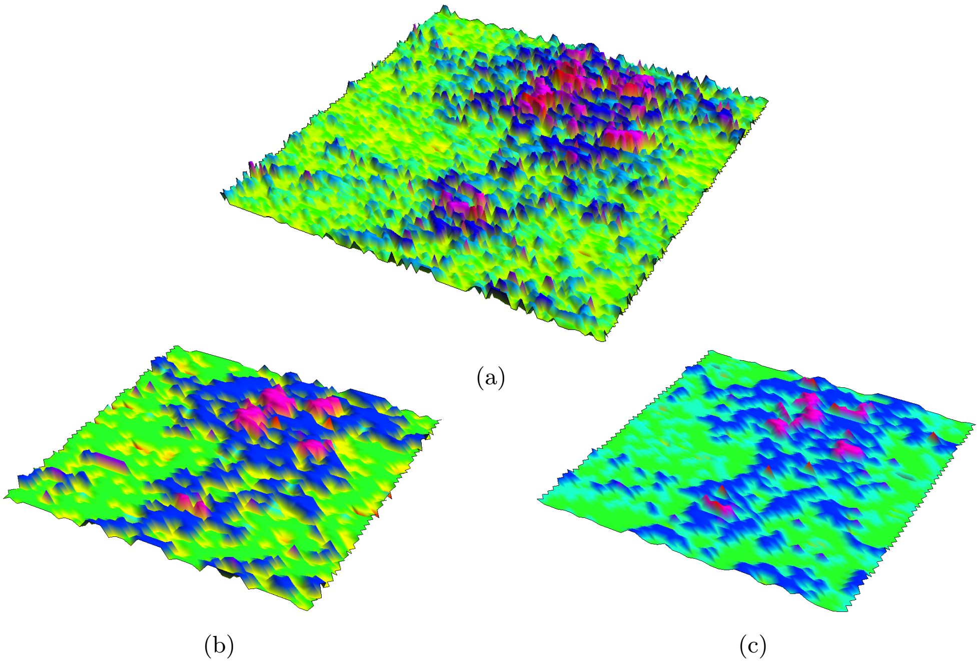


Figure 5.6: *Effects of the coarse graining procedure. Figure 5.6(a) shows the heights of the original lattice, figures 5.6(b) and 5.6(c) are the two renormalized lattices. Notice how they reproduce the general features of the original lattice, albeit both differently. Only one of the four repeating quadrants is shown.*

denoted by the red lines in figure 5.5(a), both renormalized lattices are not influenced by this defect. See figure 5.7(b).

Since the renormalization is performed using both placements of the renormalized lattice, after which the four renormalized lattices are averaged over, such an excitation of four vertices in the original lattice is expected to on average translate into a single asymmetric vertex state per renormalized lattice. The number of vertices is reduced by four in the coarse graining, hence the relative number of excited vertices due to such a single defect should remain constant under renormalization.

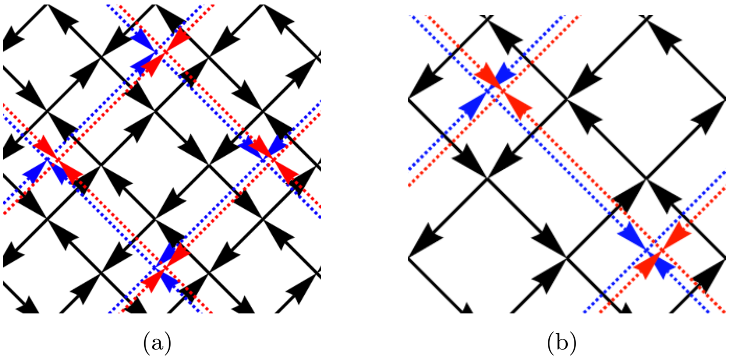


Figure 5.7: *Effect of the coarse graining procedure on the smallest possible excitation. In figure 5.7(a) the blue lattice captures the excitation, the red lattice remains undisturbed. In figure 5.7(b) both renormalized lattices do not reproduce the excitation, both are in the ground state.*

5.3 Application

5.3.1 Probability of Two Vertices to be in the Same Ground State Region

In order to apply the coarse graining prescription developed above, it is necessary to determine an observable that can be measured both before and after renormalization. The observable we focus on is the probability of two different vertices separated by a distance r to be in the same ground state region. This is denoted by $P_{\text{sr}}(r)$, where the sr stands for 'same region'. The probability $P_{\text{sr}}(r)$ is closely related to the typical behaviour around the phase transition, as shown in figure 2.5 on page 14.

The value of P_{sr} was determined by averaging over all vertices the function $\delta(i,j)$, defined by

$$\delta(i,j)\begin{cases}\text{takes no value,} & \text{if both } i \text{ and } j \text{ are not part of a ground state region,}\\ 1, & \text{if } i \text{ and } j \text{ are in the same ground state region,}\\ 0, & \text{if only one of } i \text{ and } j \text{ is in a ground state region,}\\ 0, & \text{if } i \text{ and } j \text{ are in different ground state regions.}\end{cases} \tag{5.8}$$

By 'takes no value' I mean it was not taken into account while taking the average. For each combination of two vertices i and j their Euclidean separation r was determined, and then per distance all $\delta(i,j)$ were averaged over, except those cases where neither i or j was a symmetric vertex. Averaged over a large number of F-model configurations at a given inverse temperature $\epsilon\beta$, this results in the function $P_{\text{sr}}(r)$. It is plotted for various inverse temperatures in figure 5.8.

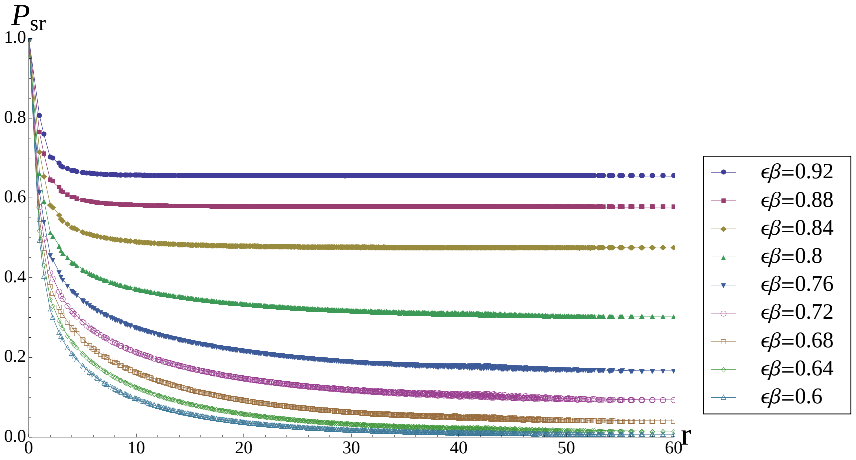


Figure 5.8: *The probability for two vertices at a separation r to be in the same ground state region, measured for various inverse temperatures on a lattice of size $L = 120$.*

Separation by a distance of more than half the lattice size is not relevant, since the periodic boundary conditions assure that two vertices are never more than half the lattice size away from each other. The distance r is expressed in the natural length scale on the lattice, where the distance between two vertices that are nearest neighbours was set to 1.

The behaviour of $P_{\text{sr}}(r)$ displays the phase transition. For low temperatures, i.e. high $\epsilon\beta$, the system is almost completely in one large ground state region. Therefore, the probability of two vertices to be in the same ground state region quickly becomes a constant significantly larger than zero. The major ground state region for $\epsilon\beta = 0.92$ covers 65.5% of the lattice.

For high temperatures, i.e. low $\epsilon\beta$, the F-model is in a disordered configuration. There are many asymmetric vertex states, and only very small ground state regions. With increasing r the probability for two vertices to be in the same ground state region quickly diminishes to zero.

In between these two states is the phase transition, for which in the previous chapter the critical inverse temperature was found to be $\epsilon\beta_\mathrm{c} = 0.79 \pm 0.02$. Indeed figure 5.8 corresponds to $\epsilon\beta_\mathrm{c} \approx 0.8$. This seems to be the value above which $P_\mathrm{sr}(r)$ goes to a constant greater than zero, and below which $P_\mathrm{sr}(r)$ eventually will decrease to zero. This suggests that the discrepancy between the measured critical temperature and the analytical value is not due to an error in the finite-size scaling.

5.3.2 Results

The same probability for two vertices to be in the same ground state region was also measured for the configurations after renormalization. It is plotted in figure 5.9. Note that the inverse temperatures in the legend refer to the inverse temperatures before renormalization. Also, while the renormalized lattice is only half the size of the original lattice, $P'_\mathrm{sr}(r)$ can only be measured up to half of this length, so that in this figure r only extends to $\frac{L}{4}$.

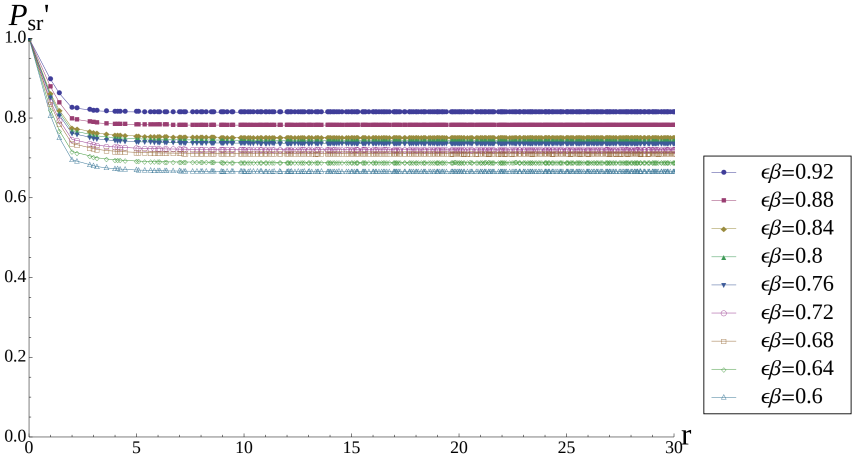


Figure 5.9: *The probability for two vertices at a separation r to be in the same ground state region after renormalization. The value of $\epsilon\beta$ refers to the value before renormalization.*

Comparing figure 5.8 and 5.9 is disappointing at least. A striking feature is the fact that all measurements after renormalization exhibit behaviour that is typical of the system at low temperatures. The probability for two vertices to be in the same ground state region for growing separation quickly becomes a constant above 0.6. This means that the renormalized lattice configurations exhibit large ground state regions. Hence, the renormalized configurations behave as low temperature configurations.

For the low temperatures ($\epsilon\beta > \epsilon\beta_\mathrm{c}$) this is not a problem, this is exactly what the renormalization is supposed to do.

For the high temperatures ($\epsilon\beta < \epsilon\beta_\mathrm{c}$) this is a problem. Under renormalization configurations with a temperature above the critical temperature should obtain a renormalized temperature that is even higher. This is further discussed in the next section.

5.4 Discussion

A comparison of figure 5.8 and 5.9 shows that the renormalized system behaves as if the temperature is much lower than in the original configuration. Even high temperatures, far above the phase transition after renormalization result in a $P'_{\mathrm{sr}}(r)$ that corresponds to a temperature far below the critical temperature.

This behaviour prevents any measurement of the critical exponents through the parameter flow around the critical point. Since high temperatures under renormalization become temperatures below the critical point, the critical point does not show in the parameter flow under renormalization, and no coefficients can be measured using this technique. In other words, the high temperature fixed point is not fixed.

The coarse graining procedure flattens the height function too much. Too many regions are fused together in large ground state regions that signify low temperatures. It must be noted that division by 2 is a rather crude measure, from a height perspective. The height-height correlation function (eq. (2.15)) only diverges logarithmically in r for temperatures above the phase transition. Therefore, the height differences do not grow very quickly, and they are easily diminished by a procedure like this. A possible solution to this would be to study significantly larger lattices, on which the height differences by nature are larger.

As the height differences become larger, the probability that they are not reflected in the renormalized configuration becomes smaller. Of course the larger height differences only consist of a series of height differences of 1, but the renormalization will less likely skip a few of these after each other.

The normal consumer grade laptop on which the simulations were performed was not able to produce statistically relevant results for much larger lattices within reasonable time of computation. The simulation of larger lattices would need a better computer, quicker algorithms, or a more efficient implementation of the algorithms used in this research. It is not unrealistic to expect that significant progress can be made on both the first and last factor without too much trouble.

Another possible explanation of the way in which the renormalization produces configurations of much lower temperature than desired could be that it is not suited towards handling small excitations. Because in some of the single-plaquette excitations the excited vertices do not lie along a central plaquette of any of the renormalized lattices, not all small excitations on a ground state region are translated into one of the renormalized lattices. Figure 5.10 shows an example.

Excitations that do not show up in the renormalized configuration do add to the temperature of the system before renormalization, whilst the renormalized system has a lower temperature since the excitations are left wiped out in the renormalization. However, this can also be said of the renormalization of the Ising model by means of a majority rule below the critical temperature. In that procedure the coarse graining prescription also tends to leave out the excitations relative to the ground state.

The mechanism of figure 5.10 contributes to the lowering of temperatures under renormalization. This should be a topic of further research. Possibly more choices for the placement of the renormalized lattice should be added and averaged over, possibly this effect only sig-

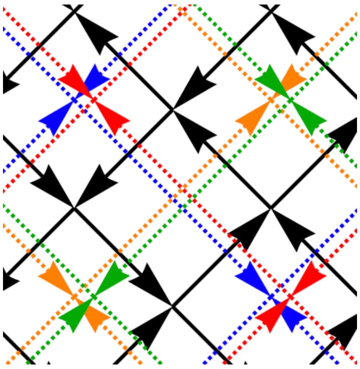


Figure 5.10: *An excitation above the ground state that is not reproduced in any of the four renormalized lattices.*

nificantly contributes to producing low temperatures under renormalization from states that originally already were in a temperature below the critical temperature. Since the high temperature configurations contain the most excitations that are possibly lost in the renormalization, the effect is expected to be most prominent there.

Along similar lines, further research could be aimed at analysing the coarse graining procedure further on the level of the arrow configurations. In this research it was formulated and best understood through the height function, but a better insight in the coarse graining at the level of arrow configurations might also show the way to a more finely tuned real space renormalization approach.

Circumventing the problem of the low temperatures after renormalization by considering a different observable does not seem a fruitful approach. The problem is that the vertex configurations after the coarse graining procedure contain too many symmetric vertex states. Since the Hamiltonian of the F-model clearly links its basic energetic behaviour to the number of symmetric vertex states in the lattice one cannot solve the appearance of too many of these configurations by considering a different observable.

At any rate, the development of a working coarse graining prescription for the F-model is a promising result, and provides confidence that future research will be able to extract more properties of the F-model from this approach.

Chapter 6

Conclusion

6.1 Summary

The F-model has proven to be a rich and surprising topic of research.

The construction of the F-model is rather straightforward. From the two ice rules, basic demands to describe the positions of the protons in a simplified model of Ice Ih, a family of six vertex models can be built. One of these is the F-model, that shows a preference for symmetric vertex states. The F-model undergoes an infinite-order phase transition, the critical temperature of which was analytically determined by Lieb [4] to be $\epsilon\beta_{\mathrm{c}} = \log[2]$.

Another way of looking at the F-model is through the height function. This assigns a collection of heights to each arrow configuration. In this image the phase transition displays as a roughening transition. This is also crucial for the connection to one of the many equivalent models, the body centered solid-on-solid model. Other equivalent models are not specifically equivalent to the F-model, but more generally to square ice. These include the three-colouring problem, and the flow of an incompressible fluid.

In chapter 3 the algorithms used to perform simulations of the F-model were presented. Firstly, two loop algorithms can be developed from the notion that turning all arrows along a closed loop that follows arrow directions will translate configurations of the F-model into other configurations of the F-model. This leaves open two ways of creating such loops, of which the short loop algorithm is the most effective.

However, the efficiency of simulation can further be improved by also employing a second algorithm, the full lattice cluster algorithm. This is based on the three-colour representation of the F-model, that is in turn only a modification of the height function. However, the full lattice cluster algorithm lacks ergodicity. configurations with a height difference from side to side that is unequal to zero when taken modulo three cannot be handled by the cluster algorithm. This is solved by applying a combined algorithm that consists of both the cluster and short loop algorithm.

The problem with the simulational approach to a phase transition that involves a diverging correlation length is the emergence of finite-size effects. However, this problem can be turned into an advantage by employing finite-size scaling. For the Ising model, this is quite straightforward, and well known. In chapter 4 finite-size scaling was applied to the F-model by means of the rescaling $\epsilon\beta \implies \log[L]^2 \frac{\epsilon\beta - \epsilon\beta_{\mathrm{c}}}{\epsilon\beta_{\mathrm{c}}}$. The finite-size scaling returned an inverse critical temperature of $\epsilon\beta_{\mathrm{c}} = 0.79 \pm 0.02$. This is quite different from the analytical value.

Possible explanations of this lie in programming errors, fundamental errors in the algorithm, and errors in the applied rescaling. However, there are valid arguments against all these explanations.

Chapter 5 dealt with real space renormalization. The core idea of this technique is to coarse grain the state of a physical system into a smaller configuration, containing less degrees of freedom, and then considering the result a state of a similar system, albeit with slightly different parameters. For the Ising model this can easily be done by replacing blocks of spins by one spin each. From the resulting flow in parameter space the critical coefficients can be obtained.

For the F-model real space renormalization turns out to be less straightforward. The usual coarse graining methods do not work because the F-model does not allow any two vertex states to be neighbours. Therefore, such local renormalization is not possible. We did manage to develop a coarse graining algorithm, based on the height function introduced earlier. This coarse graining algorithm translates any configuration of the F-model into a smaller configuration that is also an allowed configuration of the F-model. It also reproduces the general features of the original configuration in the renormalized configuration.

The behaviour under renormalization was discussed by means of the probability of two vertices to be in the same ground state region as a function of their separation. Even before renormalization these measurements confirm the critical temperature found in finite-size scaling, again suggesting the deviation from the analytical value is not due to a mistake in the theoretical basis of the finite-size scaling.

The behaviour under renormalization remains problematic. The renormalized configurations that stem from an original configuration at temperatures above the critical value show behaviour corresponding to very low temperatures, which means that the high-temperature fixed point is not fixed. The watershed that should take place at the critical temperature, from which the temperature should always flow away under renormalization, was not found. Therefore it was not possible to determine critical coefficients from the renormalization.

6.2 Outlook

The results obtained in chapter 4 were already discussed section 4.3, those of chapter 5 were discussed in section 5.4. The questions posed there could very well be the subject of further research.

The finite-size scaling obtained an inverse critical temperature ($\epsilon\beta_{\mathrm{c}} = 0.79 \pm 0.02$) that is quite different from the analytical value ($\epsilon\beta_{\mathrm{c}} = \log[2] \approx 0.693$). This calls for an explanation, especially when one considers that the graph of $P_{\mathrm{sr}}(r)$, figure 5.8, suggests a similar critical inverse temperature around $\epsilon\beta_{\mathrm{c}} = 0.8$. Such an explanation could be a goal of further research, in which a different way of measuring β_{c} yet using the same algorithms could shed some light on the question whether this critical temperature is inherently bound to these algorithms.

Further development is also possible on the renormalization procedure. In its current form it does not function well enough to measure critical coefficients. However, there are various ways in which one could try to improve this. One possibility is to consider significantly larger lattices. This requires improvements in either the algorithms, their implementation, or the computing power, but this does not seem infeasible.

Another way would be to further study the way in which the coarse graining prescription works on the level of the arrows. A more theoretical and mathematical approach to this might explain the tendency to create symmetric vertex states on the level of arrows, and therefore suggest a method for improvement. Moreover, the way the coarse graining prescription treats small excitations could be inspected more closely.

Lastly, the thought formulated in the introduction remains. If the techniques employed in this research are developed further they could possibly also be used on lattices including defects. This would be a significant improvement compared to the analytical approach, that can only deal with perfect lattices.

Acknowledgements

'No man is an island' [52], and neither is any thesis. Of course I am very grateful to various people who were of great assistance in realizing this thesis.

First and foremost, I would like to thank my supervisor, Prof. dr. Gerard T. Barkema. Not only was his help indispensable in determining the crucial directions of research, but also his humour and unwavering optimism about the results and the way of getting there were very contagious. Moreover, his great accessibility was instrumental in keeping up the pace of work.

Secondly, I would like to thank my fellow students in the students room, whose company made the little victories in research much greater, and the setbacks much easier to bear. I extend special thanks to Gertjan for his great ability to solve even the most obscure of bugs in any computer related area, Jules for his great reservoir of knowledge and solutions to any problem encountered when doing research in the natural sciences, Quirine for the many fruitful discussions about our respective projects, and Béatrice for her ever cheerful presence, even and especially in her times of crisis.

References

[1] F. Rys, Helv. Phys. Acta **36**, 537 (1963).

[2] J. D. Bernal and R. H. Fowler, J. Chem. Phys. **1**, 515 (1933).

[3] L. Pauling, J. Am. Chem. Soc. **57**, 2680 (1935).

[4] E. H. Lieb, Phys. Rev. Lett. **18**, 1046 (1967).

[5] E. H. Lieb and F. Y. Wu, *Two-dimensional Ferroelectric Models*, edited by C. Domb and M. Green, Phase Transitions and Critical Phenomena, Vol. 1 (Academic Press, London, 1972) pp. 331 – 490.

[6] M. L. Glasser, D. B. Abraham, and E. H. Lieb, J. Math. Phys. **13**, 887 (1972).

[7] H. J. Brascamp, H. Kunz, and F. Y. Wu, J. Math. Phys. **14**, 1927 (1973).

[8] R. Baxter, *Exactly Solved Models in Statistical Mechanics* (Academic Press, London, 1990) pp. 127–201.

[9] R. Baxter, J. Stat. Phys. **116**, 43 (2004).

[10] A. Rahman and F. H. Stillinger, J. Chem. Phys. **57**, 4009 (1972).

[11] J.-S. Wang, R. H. Swendsen, and R. Kotecký, Phys. Rev. B **42**, 2465 (1990).

[12] G. T. Barkema and M. E. J. Newman, Phys. Rev. E **57**, 1155 (1998).

[13] H. van Beijeren, Phys. Rev. Lett. **38**, 993 (1977).

[14] R. Youngblood, J. D. Axe, and B. M. McCoy, Phys. Rev. B **21**, 5212 (1980).

[15] J. M. Kosterlitz and D. J. Thouless, J. Phys. C: Solid State Phys. **6**, 1181 (1973).

[16] P. V. Hobbs, *Ice Physics* (Clarendon Press, Oxford, 1974).

[17] E. H. Lieb, Phys. Rev. **162**, 162 (1967).

[18] W. F. Giauque and M. F. Ashley, Phys. Rev. **43**, 81 (1933).

[19] W. F. Giauque and J. W. Stout, J. Am. Chem. Soc. **58**, 1144 (1936).

[20] J. F. Nagle, J. Math. Phys. **7**, 1484 (1966).

[21] J. F. Nagle, J. Math. Phys. **7**, 1492 (1966).

[22] E. H. Lieb, Phys. Rev. Lett. **19**, 108 (1967).

[23] B. Sutherland, Phys. Rev. Lett. **19**, 103 (1967).

[24] J. C. Slater, J. Chem. Phys. **9**, 16 (1941).

[25] J. Nagle, Commun. Math. Phys. **13**, 62 (1969).

[26] M. Weigel and W. Janke, J. Phys. A: Math. Gen. **38**, 7067 (2005).

[27] W. J. Shugard, J. D. Weeks, and G. H. Gilmer, Phys. Rev. Lett. **41**, 1399 (1978).

[28] E. Luijten, H. van Beijeren, and H. W. J. Blöte, Phys. Rev. Lett. **73**, 456 (1994).

[29] F. S. Rys, Surf. Sci. **178**, 419 (1986).

[30] S. Mitra, *Renormalization Group Studies of Six Vertex Models*, Master's thesis, Utrecht University (1999), arXiv:physics/9910031v2 [physics.class-ph].

[31] R. Youngblood, J. Axe, and B. McCoy (1979) report numbers BNL-26196; CONF-790487-1, online available at
http://www.osti.gov/bridge/product.biblio.jsp?osti_id=6146352.

[32] A. Lenard, J. Math. Phys. **2**, 682 (1961).

[33] P. Kundu and I. Cohen, *Fluid Mechanics* (Academic Press, London, 2010).

[34] R. Raghavan, C. Henley, and S. Arouh, J. Stat. Phys. **86**, 517 (1997).

[35] M. Newman and G. Barkema, *Monte Carlo Methods in Statistical Physics* (Clarendon Press, Oxford, 2009).

[36] H. Saleur, Nucl. Phys. B. **360**, 219 (1991).

[37] A. Yanagawa and J. Nagle, Chem. Phys. **43**, 329 (1979).

[38] G. T. Barkema and J. de Boer, J. Chem. Pyys. **99**, 2059 (1993).

[39] L. Kadanoff, Physics **2**, 263 (1966).

[40] K. G. Wilson and J. Kogut, Physics Reports **12**, 75 (1974).

[41] K. G. Wilson, Phys. Rev. B **4**, 3174 & 3184 (1971).

[42] M. E. Fisher, Rev. Mod. Phys. **46**, 597 (1974).

[43] S.-K. Ma, Phys. Rev. Lett. **37**, 461 (1976).

[44] R. H. Swendsen, Phys. Rev. Lett. **42**, 859 (1979).

[45] J. Binney, N. Dowrick, A. Fisher, and M. Newman, *The Theory of Critical Phenomena: An Introduction to the Renormalization Group*, Oxford Science Publications (Clarendon Press, Oxford, 1992).

[46] J. Cardy, *Scaling and Renormalization in Statistical Physics*, Cambridge Lecture Notes in Physics (Cambridge University Press, Cambridge, 1996).

[47] N. Goldenfeld, *Lectures on phase transitions and the renormalization group*, Frontiers in physics (Addison-Wesley, Boston, 1992).

[48] W. McComb, *Renormalization Methods: A Guide For Beginners* (Oxford University Press, Oxford, 2008).

[49] G. Barkema, "Lecture notes on renormalization," (2011), unpublished, obtained from the author.

[50] R. H. Swendsen, Phys. Rev. B **20**, 2080 (1979).

[51] H. W. J. Blöte and R. H. Swendsen, Phys. Rev. B **20**, 2077 (1979).

[52] J. Donne, *Devotions upon Emergent Occasions* (A.M. for Thomas Iones, London, 1624).

Appendix: A more rigorous treatment of the height function

This appendix aims to prove the single-valuedness of the height function described in section 2.3 with more mathematical rigour. Consider the following two theorems:

Theorem 6.1. *The total of all height differences in a loop around an arbitrary number of vertices adds up to zero.*

Theorem 6.2. *The height difference between two points on the lattice does not depend on the path taken in between the points.*

The short-cut way to prove both theorems would be to notice that the arrows represent a divergence-free field. Therefore it is also conservative. Hence, it is path-independent. However, let us prove this more explicitly.

Proof of theorem 6.1. Consider the function h_F that is the sum of all height steps when walking over path F. For F wrapping around one vertex $h_F = 0$. This can explicitly be verified for all six vertices, in the same way as was done in figure 2.6 on page 15. Hence for G also a path around a single vertex, $h_G = 0$. Now consider figure 6.1.

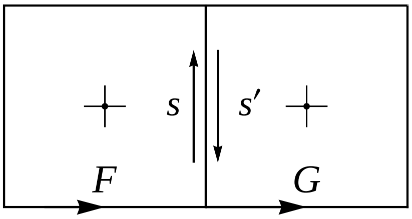


Figure 6.1: *Illustration of the proof of theorem 6.1. The two loops F and G each wrap around a single vertex. They share one line segment, $S \subset F$, $S' \subset G$, and S' is just S in the other direction.*

The paths F and G both wrap around a single vertex. Moreover, the section s is a part of F, and s' is a part of G. The only difference between s and s' is their direction. Hence $h_s = -h_{s'}$. The height difference when walking along the big loop around both vertices is given by

$$h_{\substack{\text{Loop around}\\\text{both vertices}}} = h_F - h_s + h_G - h_{s'} = h_F + h_G - h_s + h_s = h_F + h_G = 0. \tag{6.1}$$

Hence, for the loop around both vertices, leaving out the middle part, the overall height difference must still equal zero. Since the addition of further loops around a single vertex can be done in the same way, no height difference can exist even when the loop encompasses any number of vertices. □

Proof of theorem 6.2. Consider points x and y on a six vertex lattice and a loop that runs through both points. K is the path on the loop oriented clockwise. The loop L is the same, but oriented counter-clockwise. See figure 6.2.

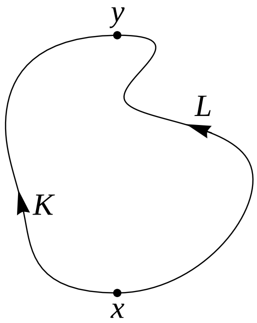


Figure 6.2: *Illustration of the proof of theorem 6.2. x and y are two points on the lattice, K and L are the two paths that lie on a random loop through both points. They are oriented in opposite directions.*

From the last theorem we know $h_K = h_L = 0$. Let $h(x \to y, K)$ be the total height difference encountered when walking from x to y along path K.

Now we know

$$h(x \to y, L) = -h(y \to x, K) = h(x \to y, K). \tag{6.2}$$

Here the first equality is due to the difference in directions, even though both sides describe a flow on the same path. The second one follows from $h_K = 0 = h(x \to y, K) + h(y \to x, K)$. The above equation shows that $h(x \to y, L) = h(x \to y, K)$, hence the height difference between x and y does not depend on the path taken between them. □