Computational approach to scaling and criticality in planar Ising models

Mikhail Dudalev

January 2012

A thesis submitted for the degree of Doctor of Philosophy
of the Australian National University

Declaration

This thesis is an account of research undertaken in the Department of Theoretical Physics within the Research School of Physics and Engineering at the Australian National University between June 2007 and July 2011 while I was enrolled for the Doctor of Philosophy degree.

I hereby declare that, unless specifically stated otherwise, the material presented within this thesis is my own. None of the work presented here has ever been submitted for any degree at this or any other institution of learning.

Mikhail Dudalev
10 January, 2012

Acknowledgements

First and foremost, I would like to thank my supervisors: Vladimir Mangazeev, Vladimir Bazhanov and Murray Batchelor. Without their devoted help, wise advice and supervision this work would not be possible at all.

During my term in the department, I enjoyed conversations and collaboration with many science-dedicated people, including Jaques Perk, Helen Au-Yang, Yaoban Chan, Sergey Sergeev, Mukunda Das and Xi-Wen Guan; as well as other students: Jen-Yee Lee, Yves Barmaz, Joshua Garretson, Andrew Kels, Imam Alam, Matthew McGann and Jonny Yi. I also would like to thank Tomotoshi Nishino and Guifré Vidal for their insightful comments at StatPhys24.

Since computers were the main investigation tools in my research, I'm obliged to praise the people who answered a lot of programming-related questions, namely Andrey Bliznyuk and Vadim Zborovskii.

Among other people to thank are my friends who survived my lame attempts to explain what I study, Russian communities of Canberra and Sydney, especially Kate Kiseeva and Marina Lobastov for persuading me to finish, and of course my parents, Yuri and Galina, for their constant moral support.

Finally, I thank the Australian National University and the Australian Research Council for funding my research.

Abstract

In this thesis, we study the critical behaviour of the two-dimensional Ising model on the regular lattices. Using the numerical solution of the model on the square, triangular and honeycomb lattices we compute the universal scaling function, which turns out to be identical on each of the lattices, in addition to being identical to the scaling function of the Ising Field Theory, computed previously by Fonseca and Zamolodchikov.

To cope with the lattice contributions we carefully examined series expansions of the lattice free energy derivatives. We included the non-scaling regular part of the free energy as well as non-linear Aharony-Fisher scaling fields, which all have non-universal expansions. Using as many of the previously known exact results as possible, we were able to fit the unknown coefficients of the scaling function expansion and obtain some non-universal coefficients. In contrast to the IFT approach of Fonseca and Zamolodchikov, all coefficients were obtained independently from separate datasets, without using dispersion relations.

These results show that the Scaling and Universality hypotheses, with the help of the Aharony-Fisher corrections, hold on the lattice to very high precision and so there should be no doubt of their validity.

For all numerical computations we used the Corner Transfer Matrix Renormalisation Group (CTMRG) algorithm, introduced by Nishino and Okunishi. The algorithm combines Baxter's variational approach (which gives Corner Transfer Matrix (CTM) equations), and White's Density Matrix Renormalisation Group (DMRG) method to solve the CTM equations efficiently. It was shown that given sufficient distance from the critical point, the algorithmic precision is exceptionally good and is unlikely to be exceeded with any other general algorithm using the same amount of numerical computations.

While performing tests we also confirmed several critical parameters of the three-state Ising and Blume-Capel models, although no extra precision was gained, compared to previous results from other methods. In addition to the results pre-

sented here, we produced an efficient and reusable implementation of the CTMRG algorithm, which after minor modifications could be used for a variety of lattice models, such as the Kashiwara-Miwa and the chiral Potts models.

Citations to previously published papers

Publications related to this thesis are as follows:

- V. V. Mangazeev, M. T. Batchelor, V. V. Bazhanov, and M. Yu. Dudalev. Variational approach to the scaling function of the 2D Ising mode in a magnetic field. *J. Phys. A*, **42** 042005. (2009)
- V. V. Mangazeev, M. Yu. Dudalev, V. V. Bazhanov, and M. T. Batchelor. Scaling and universality in the two-dimensional Ising model with a magnetic field. *Phys. Rev. E*, **81** 060105. (2010)
- M. Yu. Dudalev, V. V. Mangazeev, V. V. Bazhanov, and M. T. Batchelor. Scaling function of the honeycomb lattice Ising model. *In preparation.*

Contents

Acknowledgements **v**

Abstract **vii**

Citations to previously published papers **ix**

Introduction **1**

1 Background **7**
1.1 2D Ising Model 7
1.1.1 Full Solution in Zero Field 10
1.2 Baxter's Variational Principle 15
1.2.1 Main Equations 19

2 Corner Transfer Matrix Algorithm **25**
2.1 Renormalisation Group 25
2.1.1 Density Matrix Renormalisation Group 26
2.1.2 Corner Transfer Matrix Renormalisation Group 28
2.2 Variations of the Algorithm 32
2.2.1 Square Lattice 32
2.2.2 Triangular lattice 35
2.2.3 Honeycomb lattice 37
2.3 Other Related Algorithms 45

3 Scaling of the Ising Model **47**
3.1 Scaling and Universality Hypotheses 47
3.1.1 Critical Indices 48
3.1.2 Ising Model Scaling Function 50
3.1.3 Universality Hypothesis 51
3.2 Ising Field Theory 52

3.2.1 Scaling Function of the IFT . . . 53
3.3 Lattice Corrections to Scaling . . . 57
3.3.1 Regular Part of the Free Energy . . . 58
3.3.2 Aharony-Fisher Non-Linear Lattice Fields . . . 60
3.3.3 Irrelevant Operators . . . 63

4 Numerical Results 65
4.1 Details of the Calculations . . . 65
4.1.1 Convergence speed and precision . . . 70
4.1.2 CTM structure . . . 72
4.2 Fitting the expansion coefficients . . . 75
4.2.1 Complete Series . . . 76
4.2.2 Regression procedure . . . 79
4.2.3 Non-universal coefficients . . . 82
4.2.4 Irrelevant Operators . . . 83
4.3 The Scaling Function . . . 83

5 Other Models 89
5.1 Spin-1 Ising model . . . 90
5.1.1 Transition Temperature and the Critical Exponent . . . 91
5.2 Blume-Capel Model . . . 93

Conclusion 94

Implementation Details 98

A Implementation Details 99
A.1 Program Structure . . . 99
A.1.1 Variable names . . . 100
A.1.2 File System Data Formats . . . 101
A.1.3 Subprograms . . . 102
A.1.4 Global Parameters . . . 104
A.1.5 QUAD Precision Notes . . . 106
A.1.6 Source Code Organisation and Program Building . . . 107
A.2 User Manual . . . 110
A.3 Known Issues and Limitations . . . 112

Bibliography 115

Introduction

Condensed matter physics is one of the most widely studied areas of physics. The well known public preprint Arxiv has an umbrella section "cond-mat" that has a larger number of publications per month than any other section. This seems to be quite natural, because this branch of physics enjoys considerable applicability in modern technology; one could say it is directly commercialisable. Just as quantum mechanics serves as a basis for all other quantum-related physics, statistical mechanics forms the foundation of condensed matter physics. The elementary laws governing small elements of the system multiplied by the huge number of those elements creates plausible explanations for the majority of the effects we see in experiments. Statistical mechanics also shows a deep interconnection with quantum field theory, both being mutual sources of inspiration and techniques during the last few decades.

One of the simplest models in statistical mechanics is the Ising model. It was originally introduced as a model of a phase transition in magnets, i.e. the Curie point. While viewing the magnet as just a combination of small magnets aligned along a single axis and interacting only with their nearest-neighbours could be viewed as an oversimplification, it has nevertheless revealed rather rich physics, and in fact can be seen as more correct model than a naive ensemble of magnets due to quantum effects. When it was originally proposed by Lenz [1] and solved for the 1D case by Ising [2], rich physics was not seen because for the 1D case there is no phase transition in the model, and the free energy is always a regular function except at zero temperature. Overgeneralisation of this result even led to the theories that still reside in some text books, namely Landau [3] theory, that most phase transitions are not represented by any singularities in the free energy, but are just some change of the internal arrangements or symmetry.

The above mentioned theory was clearly seen to be wrong when the 2D case was solved for zero external field by Lars Onsager in 1944 [4]. The solution clearly showed that the phase transition is an actual singularity and that the

specific heat becomes infinite at the critical point. Onsager's work is also credited with creating a branch inside statistical mechanics, devoted to the exactly solved models, which in turn catalysed development of many new areas of mathematics, such as quantum algebra.

But despite numerous attempts there is still no exact solution for the non-zero field 2D Ising Model. There is no exact solution for the three-dimensional Ising model as well, but in dimensions more then three the Ising model could be treated very well using a mean-field approximation, although no exact solution exists. There are a lot of other 2D models which have been solved exactly (like 6-vertex [5] and 8-vertex [6] models, the Kashiwara-Miwa [7, 8] model and many others). Naturally, all these models are exceptions, since general lattice models can not be solved exactly. It appears though that the most interesting things are happening in the vicinity of the critical point. One such thing is scale invariance, which also could help to get an approximate solution of the model near the point where it works. Scaling is an old and obvious concept, that is especially easy to understand in mechanics. The rescaled system has the same properties as the original system, and differs only by some scaling factor. Scale invariance means that some system's physics is unchanged by rescaling. Measured values, if they are not dimensionless, of course are also scaled, but the whole physics of the system, and dimensionless combinations of observable values stay the same. Sufficiently close to the critical point the correlation length becomes infinitely large and can not be a natural scale any more. Thus the free energy becomes a homogeneous function of the model parameters, or, in another words,a function only of the dimensionless combinations of the model parameters.

The first consistent application of scaling invariance (or strictly speaking a Renormalisation Group theory) came to quantum field theory with works of Gell-Mann and Low [9], and a little bit later was introduced to statistical mechanics by Fisher [10] in the form of critical exponents, i.e. dimensions of the model's parameters and observable values near the critical point. Fisher was one of the people who proposed the scaling and universality hypotheses. The first hypothesis states that around the critical point a singular part of the free energy is expressed via a scaling function which depends on the dimensionless combinations of the model parameters only. The universality hypothesis states that all models are split in equivalence classes depending on dimensionality and symmetry, and share the same scaling function. There is also a belief that there are a lot of interesting models that belong to the Ising universality class.

While there is a lot of evidence that these hypotheses are true, strictly speak-

ing there is no proof or precise demonstration even for the Ising model itself, because of the absence of a solution with sufficient precision (let alone an exact solution) for $H \neq 0$.

Nevertheless, the hypotheses were accepted, used and developed, e.g. by Kadanoff and Wilson. Works of the latter brought the statistical mechanics variant of scale invariance to quantum field theory. Generalisation of the scale invariance to the local scale transformation in 2D gave a brilliant and powerful theory called Conformal Field Theory [11], which found widespread use in many areas of theoretical physics. It also describes the critical limit of the lattice Ising Model allowing us to obtain its approximate solution around the critical point, and ultimately to verify the scaling and universality hypotheses.

Besides the family of exact solutions of the lattice statistical models, in the 70's a related field of research began to flourish. Starting from the quantum inverse scattering method [12, 13], a branch of quantum field theory dedicated to exact solutions started to develop. One of the key ingredients of the exactly solved QFT models is the Yang-Baxter equation. Nowadays there are also many exactly solved models related to string theory, AdS/CFT models and many others. Most of them use methods originally developed in statistical mechanics.

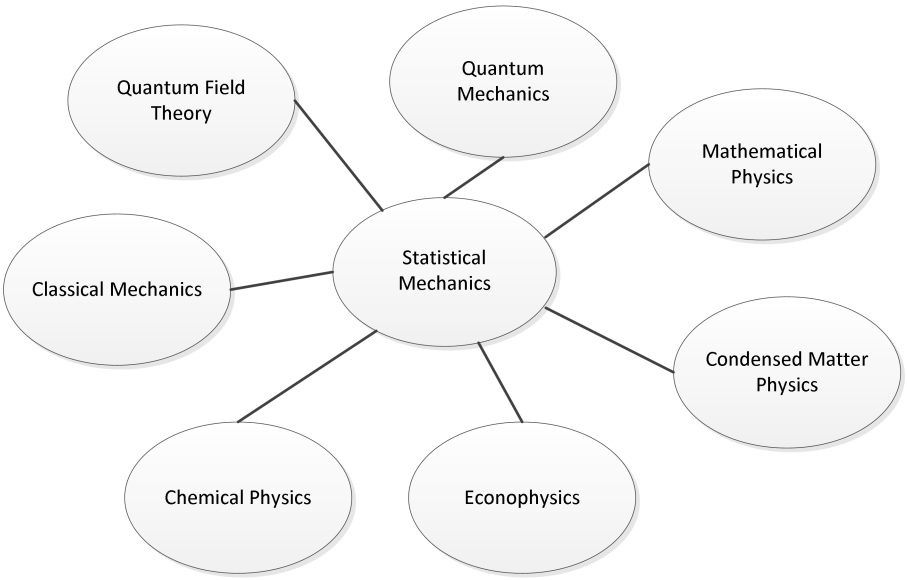

Figure 1: Relation of statistical mechanics to the other branches of physics.

The motivation of the present work was to convincingly demonstrate the validity of the scaling and universality hypotheses using high-precision numerical methods, which in turn, ironically, descend from the application of scaling invariance. A short description of the chapters now follows.

In the first part of the Chapter 1 we describe the 2D Ising model, its solution for the $H = 0$ case, and other exact results that are relevant to obtaining the scaling function, such as the series for the magnetic susceptibility. The second part is devoted to explaining what Baxter's variational approach is, and how very important objects called Corner Transfer Matrices emerge in this approach. The approach gives a system of matrix equations with infinite matrices. While it is not entirely rigorous, it is possible to show that the truncated form of the equations with finite matrices is a good approximation to the infinite case.

There are many ways to solve a system of non-linear equations, such as the Newton-Raphson algorithm; some of them are well-suited to the matrix system. But for Baxter's equation a special iterative method exists. It not only solves the equations, but gives them an extra physical sense; connecting them with White's Real Space Renormalisation Group [14], originally proposed to compute a density matrix in quantum models. Developed by Nishino and Okunishi [15, 16], the method seems to be the most optimal way of solving Baxter's equations. For instance, for the Ising model it always converges (that would be a miracle for any general iterative non-linear method). In Chapter 2 we first describe in detail the White and Nishino methods. Then various generalisations of the algorithm to the non-square lattices are made, such as regular triangular and honeycomb lattices, as well as for lattices with symmetry group less than the full group of symmetries for such a lattice, e.g. the rectangular lattice with different bond strengths for horizontal and vertical bonds.

We also provide practical details of the actual algorithm implementation, the number of iterative steps, and practical observations that helped the research. For example we noticed that the structure of the Corner Transfer Matrix for $H \neq 0$ bears only a superficial resemblance to the exact $H = 0$ matrix and essentially only follows the universal asymptotics, but does not have any degenerate levels that are believed to be a sign of integrability of the system.

Technical details of all computer codes plus a short user manual for them constitutes Appendix A. The actual program is a combination of small subprograms, that are written in Fortran (the performance-critical part) and Python (the control part). Most of the data flow between these subprograms is saved to disk, so it is very easy to do the computation on cluster machines using almost no parallelisation techniques, but with just a simple distribution of the points in parameter space among different nodes of the cluster.

Besides the above mentioned algorithms, at the beginning of our research we tried to use a couple of other algorithms without much success. At the end of

Chapter 2 we outline these algorithms together with possible reasons for their failure.

Without any additional knowledge of the scaling behaviour of the Ising model free energy it is impossible to reach high precision while computing the scaling function. While it is possible to use only numerical results to get the scaling function, it is much more useful to employ as many exact results as possible. Thus it is worth gathering as much additional information about the non-zero field Ising model as possible and then study carefully the scaling behaviour of the model. This is also motivated by the actual structure of the Ising Model scaling function, which has not just a power law singularity but has an additional logarithmic singularity, which requires special treatment to extract the scaling function. The zero-field solution is given in Chapter 1; there are also several notable results for the non-zero field case which are presented in Chapter 3. Some of them relate to not just the lattice Ising model, but also to the so called Ising Field Theory (IFT), a Conformal Field Theory with central charge $\frac{1}{2}$ perturbed by two operators with coupling constants corresponding to the temperature and magnetic fields. This theory was thoroughly examined by Fonseca and Zamolodchikov [17]. Zero coupling to the magnetic field operator gives the singular part of the Onsager solution, while zero coupling to the thermal operator gives Zamolodchikov's E_8 theory [18], an exact solution for the critical isotherm of the IFT. It is exact in the IFT only, nevertheless it gives two expansion coefficients of the scaling function exactly, and that helps greatly determining the next coefficients with increased precision. Finally, Vicari et al. [19] showed how to link all the pieces to the lattice theory and that it is possible to track so called irrelevant operators which essentially are higher-order deviations from the scaling theory on the lattice.

In Chapter 4 all these techniques are applied to the numerical data obtained using algorithms from Chapter 2. Three regular lattices are studied thoroughly: the square, triangular and honeycomb lattices. While as expected all lattice values, magnetisation and internal energy have different values, all non-universal functions (Aharony-Fisher scaling fields and a regular part of the free energy) are different, but the function that lies at the heart of these functions, that is the scaling function, is absolutely the same for the three lattices, and coincides with the function obtained from IFT. This equality is demonstrated with unprecedented precision, that moves beyond any doubt the validity of the universality hypothesis. While it is very difficult to analyse high-order contributions of irrelevant operators, we also are able to partially confirm conjectures about the first irrelevant operators, i.e. their contribution to the fourth next to the leading order

in zero-field susceptibility of the square lattice Ising model.

In the final chapter, Chapter 5, we describe additional lattice models, namely the three-state Ising model and Blume-Capel model and present some basic results obtained with a simple extension of the algorithm to models with more than two states per spin. Not a lot of new results are described there, but we independently confirmed some earlier results (that were obtained by different methods specific to the models) using CTMRG (which is a rather general method). For models known much less well than the Ising model that is of great value. Also this chapter demonstrates how important it is to know some exact results in order to get the most from the numerical data; the precision of calculations was quite high, but because of the absence of knowledge of the exact critical temperature, the high precision data yielded a precision for the critical parameters and exponents that provides very little improvement over pre-existing results.

The Conclusion outlines two important things. The first are missing parts of the current research, plus a list of what could be usefully included in the current work or what was started, but wasn't finished or included. The second part contains the most promising directions for future research, albeit the plausibility of all the proposed research ideas needs careful assessment.

Chapter 1

Background

1.1 2D Ising Model

The Ising Model is one of the most important models in statistical mechanics. In a few words it can be described as a model with pairwise interaction depending on variables (called "spins") situated at vertices of some graph and having only two values ± 1, with energy of interaction being proportional to the product of adjacent spins. An optional external field H can be added, then each spin contributes an energy proportional to its value times the field strength. There is a wealth of research involving the Ising model on different types of regular lattices or complicated graphs, with applications ranging from general relativity to neural networks. In statistical mechanics one is mostly interested in a very large or infinite model on a regular lattice and in equilibrium. By a "regular lattice" in 2D we mean a lattice consisting of regular polygons: squares, triangles or hexagons. There are other interesting cases that could be solved by the methods used in this thesis, but we don't consider them here.

In higher dimensions there are some interesting results for hyper-cubic lattices, with the 3D case being arguably one of the most interesting, with many hypotheses and methods struggling to solve the $H = 0$ case. But there are very few solved 3D models and most of the results are numerical and quite inaccurate in comparison with the 2D case.

On the other hand the theory of exactly solvable models is very well developed and actually started with the solution of the 2D Ising Model with zero external field. All 2D models solved to this time belong to the class of integrable models, i.e. they have an infinite number of commuting integrals of motion. These integrals are explicitly known for $H = 0$ [20].

Let's describe the 2D Ising Model on the regular lattices. We mostly use the IRF("Interaction Around the Face") representation, where the "building blocks" of the lattice are faces and variables are point-wise spins. The other representation that is often used is the vertex form, where one constructs lattices from vertices, and lattice variables are "arrows" denoting the difference between adjacent spins, whose values are implicit.

The partition function of the Ising model is

$$Z(\beta, H) = \sum_{\{\sigma_i\}} \exp\left(\beta \sum_{ij} \sigma_i \sigma_j + H \sum_i \sigma_i\right), \qquad \sigma_i = \pm 1, \tag{1.1}$$

where the second summation runs over all bonds between adjacent spin variables σ_i, the third runs over all spins, and finally the first one runs over all spin configurations. If there are N spins on the lattice, then obviously the number of all spin configurations is 2^N. Parameters β and H are the reciprocal temperature and external magnetic field, respectively.

In many texts the same partition function is written in a more "real-world" style, first introducing an explicit Hamiltonian, i.e. the energy of the lattice configuration expressed via its microscopic state

$$\mathcal{H}(H, \{\sigma_i\}) = -\sum_{ij} J\sigma_i\sigma_j - H\sum_i \sigma_i, \qquad \sigma_i = \pm 1, \tag{1.2}$$

and then plugging it into the "Boltzmann distribution"

$$Z(T, H) = \sum_{\text{states}} e^{-\frac{\mathcal{H}(H,\{\sigma_i\})}{T}}, \tag{1.3}$$

where J is usually normalised to make the critical temperature T_c equal 1.

There is no real difference between these definitions, and one should stick to the one which is more convenient. However, it turns out that when exploring the region around criticality and using expansions that come from field theory (see Chapter 3) it is better to take the form (1.1), otherwise many series expansions are more complicated than they need to be.

The macroscopic statistical properties of the system can be expressed as derivatives of the free energy per site:

$$F(\beta, H) = -\lim_{N\to\infty} \frac{\log Z(\beta, H)}{N}. \tag{1.4}$$

Here we drop the T factor from the usual textbook definitions, since it causes many of the expressions in Chapter 3 to be much more complex than they need to be. On the other and there are a few extra factors that might appear in the usual thermodynamic formulas with the free energy.

The two main thermodynamic quantities for any system with magnetic interaction are the internal energy and the magnetisation. The magnetisation is simply an average value of a spin

$$M(\beta, H) = \langle \sigma \rangle = -\frac{\partial F(\beta, H)}{\partial H}. \tag{1.5}$$

Another important physical quantity is the internal energy. It is slightly more tricky than the magnetisation, because there are several normalisations used. The usual definition is

$$U = \langle \sigma_i \sigma_{i+1} \rangle = -\frac{\partial F}{\partial \beta}, \tag{1.6}$$

where σ_i and σ_{i+1} are adjacent (connected with a bond) spins. In this definition we calculate the internal energy per lattice bond. Sometimes the internal energy is calculated per face or per spin. To track the correspondence between spins, bonds and faces, we summarise it in a small table.

Lattice	Spins	Bonds
Square	1	2
Triangular	1/2	3/2
Honeycomb	2	3

Table 1.1: The number of spins and number of bonds per face for the lattices under consideration

The coordination number (number of bonds per spin) plays a crucial role in computing the coefficients between lattice and field quantities, i.e. coefficients C_h and C_τ for different lattices. In the numerical formulas for the internal energy the normalisation depends on the type of the lattice, since the internal energy is usually counted per bond, and not per face. In the presence of the magnetic field we might also need to subtract the magnetic part of the energy from the total internal energy to get the "thermal" part of it (that is usually required).

Most of the numerical computations to obtain the scaling function involve calculation of the magnetisation and internal energy of the system. While it is possible to compute only the free energy and then use numerical derivatives to devise all other values, it is rather inefficient. One of the decisive advantages of the method used for all computations is that it produces the magnetisation and

internal energy directly. There are still cases where the free energy needs to be calculated (in addition to tests), namely the equation (4.4) below.

Critical parameters are obviously different for different lattices. The critical value β_c for the square lattice was found in ground-breaking work of Kramers and Wannier [21]. Using just duality relations it is possible to find β_c for all three main types of the lattice [20]. The critical values for the three main lattices are:

$$\beta_c^{\mathrm{sq}} = \frac{1}{2}\log(1+\sqrt{2}) \approx 0.440687\ldots, \tag{1.7}$$

$$\beta_c^{\mathrm{tr}} = \frac{1}{4}\log 3 \approx 0.274653\ldots, \tag{1.8}$$

$$\beta_c^{\mathrm{hc}} = \frac{1}{2}\log(2+\sqrt{3}) \approx 0.658479\ldots. \tag{1.9}$$

1.1.1 Full Solution in Zero Field

By "exactly solved" here we mean that the model's free energy can be written in a closed form. Since for the Ising Model only $F(\beta, 0)$ is known, we could also require some derivatives of F in respect to H as an addition to the exact solution. Currently this addition is only the first derivative, i.e. the magnetisation $M(\beta)$, but we also use results for the second derivative - the susceptibility $\chi(\beta)$.

The method of Onsager is very complicated, but many simpler solutions emerged afterwards. Notable solutions are

- Kaufman greatly simplified Onsager's solution [22].
- Houtappel solved the Ising model on the triangular and honeycomb lattices [23].
- The solution of Vdovichenko using loop counting, it is one of the simplest solutions [24, 3].

Baxter proposed at least two independent notable solutions:

- The "399th solution of Ising model" [25]; there is also a variant of this solution which even doesn't involve elliptic function parametrisation [26].
- The Corner-Transfer Matrix-based solution for the eight-vertex model [27, 28].

Free energy The full expression for the free energy on the square lattice is [4]

$$F_{\rm sq} = -\log 2 - \frac{1}{8\pi^2}\iint_0^{2\pi} d\phi_1 d\phi_2 \log\left[\cosh^2 2\beta - \sinh^2 2\beta(\cos\omega_1 + \cos\omega_2)\right], \tag{1.10}$$

and for the triangular and honeycomb lattices [23]

$$\begin{aligned} F_{\rm tr} = &-\log 2 - \frac{1}{8\pi^2}\iint_0^{2\pi} d\phi_1 d\phi_2 \log(\cosh^3 2\beta + \sinh^3 2\beta \\ &- \sinh 2\beta \cdot [\cos\phi_1 + \cos\phi_2 + \cos(\phi_1+\phi_2)]), \end{aligned} \tag{1.11a}$$

$$\begin{aligned} F_{\rm hc} = &-\log 2 - \frac{1}{16\pi^2}\iint_0^{2\pi} d\phi_1 d\phi_2 \log\frac{1}{2}(\cosh^3 2\beta + 1 \\ &- \sinh^2 2\beta \cdot [\cosh\phi_1 + \cosh\phi_2 + \cosh(\phi_1+\phi_2)]). \end{aligned} \tag{1.11b}$$

One can introduce another temperature variable $\tau(\beta)$ in which a singular part of the free energy is an even function. This form is later used to obtain series that are required for the accurate calculation of the scaling function.

$$\begin{aligned} F_{\rm sq} = &-\frac{1}{2}\log(4\sinh 2\beta) \\ &-\frac{1}{8\pi^2}\iint_0^{2\pi} d\phi_1 d\phi_2 \log(2\sqrt{1+\tau^2} - \cos\phi_1\cos\phi_2), \end{aligned} \tag{1.12a}$$

$$\begin{aligned} F_{\rm tr} = &-\frac{1}{2}\log(4\sinh 2\beta) \\ &-\frac{1}{8\pi^2}\iint_0^{2\pi} d\phi_1 d\phi_2 \log(3+\tau^2 - \cos\phi_1 - \cos\phi_2 - \cos(\phi_1+\phi_2)), \end{aligned} \tag{1.12b}$$

$$\begin{aligned} F_{\rm hc} = &-\frac{1}{2}\log(2^{3/2}\sinh 2\beta) \\ &-\frac{1}{16\pi^2}\iint_0^{2\pi} d\phi_1 d\phi_2 \log(3+\tau^2 - \cos\phi_1 - \cos\phi_2 - \cos(\phi_1+\phi_2)), \end{aligned} \tag{1.12c}$$

where $\tau(\beta)$ is different for each lattice, namely

$$\tau_{\mathrm{sq}} = \frac{1}{2}\left(\frac{1}{\sinh 2\beta} - \sinh 2\beta\right), \tag{1.13a}$$

$$\tau_{\mathrm{tr}} = \frac{1-4v+v^2}{\sqrt{2v}(1-v)} = \frac{1}{e^{\beta}\sqrt{\sinh 2\beta}} - e^{\beta}\sqrt{\sinh 2\beta}, \tag{1.13b}$$

$$\tau_{\mathrm{hc}} = \frac{1-3v^2}{v\sqrt{2(1-v^2)}}, \tag{1.13c}$$

with $v = \tanh\beta$ for all three lattices. For temperatures lower than critical ($\beta > \beta_c$) one has $\tau < 0$.

There is another natural parametrisation arising in all solutions of the model, namely parametrisation by elliptic modulus. Using this parametrisation the magnetisation is expressed in the same way for all three main lattices.

For the square lattice [4]

$$k_{\mathrm{sq}}^2 = \frac{16e^{8\beta}}{(e^{4\beta}-1)^4} = (\sqrt{(1+\tau^2)}+\tau)^4, \tag{1.14}$$

and for the regular triangular and honeycomb lattices [23]

$$k_{\mathrm{tr}}^2 = \frac{16e^{4\beta}}{(e^{4\beta}-1)^3(e^{4\beta}+3)}, \tag{1.15a}$$

$$k_{\mathrm{hc}}^2 = \frac{16e^{6\beta}(e^{6\beta}+1)(e^{2\beta}+1)^3}{(e^{4\beta}-1)^6}. \tag{1.15b}$$

For both honeycomb and triangular lattices:

$$k(\tau) = \frac{\tau^4+6\tau^2+6+\tau(\tau^2+4)^{3/2}}{2\sqrt{2\tau^2+9}}. \tag{1.16}$$

For $\beta > \beta_c$ we have $k < 1$, the critical point corresponds to $k = 1$.

These formulas looks similar and interconnected because of duality relations. Duality was first noticed by Kramers and Wannier [21] and used to determine β_c exactly, since the relations map the low-temperature square lattice model to the high temperature one, so the square lattice model is self-dual.

For the honeycomb and triangular lattices it maps the low-temperature configuration of one lattice to the high-temperature configuration of the other. With some additional correspondence between low- and high-temperature regimes on a single type of lattice, it is possible to find critical temperatures and other prop-

erties of the triangular and honeycomb lattice Ising models. The most common generalisation of this duality is the Yang-Baxter equation, which is now ubiquitous in the theory of exactly solved systems. Unfortunately, the duality holds only for $H = 0$ and cannot be used in numerical studies for $H \neq 0$.

Magnetisation While it is now believed that Onsager knew the proof of the 2D Ising Model magnetisation formula, that he published first [29], the derivation of the formula (along with other important results) was first presented by Yang [30]. It is simply

$$M = (1 - k^2)^{1/8} \sim (\beta - \beta_c)^{1/8} \qquad \text{for } \beta > \beta_c \, (T < T_c), \tag{1.17a}$$

$$M = 0 \qquad \text{for } \beta < \beta_c \, (T > T_c). \tag{1.17b}$$

A plot of the magnetisation is shown in Figure 1.1.

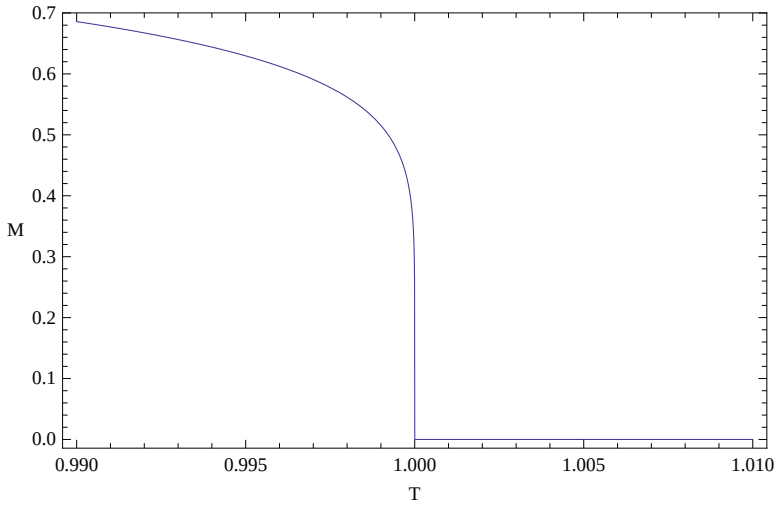

Figure 1.1: Magnetisation of the Ising model in zero field, $T = \beta_c/\beta$.

Susceptibility The susceptibility of the 2D Ising model is defined as

$$\chi = \frac{\partial M}{\partial H} = -\frac{\partial^2 F}{\partial H^2}. \tag{1.18}$$

The current situation with regard to the exact expression for the susceptibility for $H = 0$ is more complicated than with the free energy and magnetisation. While it is possible to compute the susceptibility value very precisely for any temperature and zero field using different numerical and semi-analytical methods, no closed

form is known. The first breakthrough in the subject was made by Barouch, McCoy, Tracy and Wu [31]. The most current results obtained by [32, 33] show that the susceptibility can be expressed in a semi-closed form as a sum of two series:

$$\chi_{\pm}(\tau) = C_0^{\pm}|\tau|^{-7/4}F_{\chi}(\tau) + B_{\chi}(\tau), \tag{1.19}$$

where $F_{\chi}^{\pm}(\tau)$ is a "scaling series", which is different (except for a few first terms) for $\tau > 0$ and $\tau < 0$

$$F_{\chi}(\tau) = \sum_{i=0}^{\infty} F^{\pm(i)}\tau^i, \tag{1.20}$$

and $B_{\chi}(\tau)$ is a so-called "short-distance" term, which is the same for both regimes, but has more complicated structure

$$B_{\chi}(\tau) = \sum_{q=0}^{\infty}\sum_{p=0}^{\lfloor\sqrt{q}\rfloor} b^{(p,q)}\tau^q(\log|\tau|)^p. \tag{1.21}$$

In [32] the factor $k(\tau)$ is written explicitly in $B_{\chi}(\tau)$, but in our case that is not needed. $C_0^{\pm}$ are known to very high precision.

Only the very first terms of these series are known exactly, but there are well established algorithms for computing as many terms and as high precision as required. A plot of the susceptibility is shown in Figure 1.2.

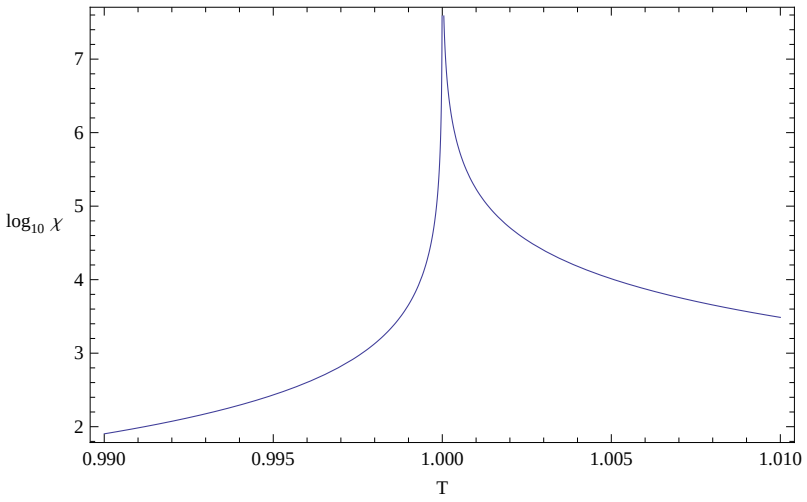

Figure 1.2: Susceptibility of the Ising Model in zero field, $T = \beta_c/\beta$.

1.2 Baxter's Variational Principle

As already emphasised, there is no exact solution for the Ising model with $H \neq 0$, so one cannot test the scaling hypothesis exactly. What one might do is to obtain a very precise, but not exact, solution and hence show that the hypothesis is correct with great accuracy.

One of the most well-known and widely used methods for solving 2D statistical models on regular (translationally-invariant in at least one direction) lattices is the Transfer Matrix method, introduced by Kramers and Wannier [21]. It is used in both analytical and numerical methods and has a solid theoretical foundation. The very essence of the method is to calculate the largest eigenvalue of the matrix built from the model's Boltzmann weights. This matrix is called the Transfer Matrix (TM) because it represents a "transfer" of the states from one lattice row to another, i.e., each matrix element equals the contribution of a piece of the lattice with given boundary conditions to the total partition function.

Consider for example a square lattice of size $m \times n$ (m columns and n rows), so that the total number of sites equals $N = mn$. Then the row-to-row transfer matrix Λ is

$$\Lambda = \prod_{i=1}^{m} w(\sigma'_i, \sigma'_{i+1}, \sigma_i, \sigma_{i+1}) \tag{1.22}$$

where the Boltzmann weight w is defined as

$$w(a, b, c, d) = \exp\left[\frac{\beta}{2}(ab + cd + ac + bd) + \frac{H}{4}(a + b + c + d)\right]. \tag{1.23}$$

The arrangement of spins a, b, c, d is shown in Figure 1.3. This is the standard spin arrangement used in this thesis. Note that only relative arrangements really matters, since the algorithms used don't distinguish between rows and columns, and we use only non-chiral models everywhere in this work, so in principle we could mirror the system and rotate it around the origin to get the same system. In some formulas left-to-right matrix multiplication maps both to a clock-wise direction in the pictures, in the other to an anti-clock-wise direction, but in general this doesn't cause any confusion.

To get the partition function one needs to compute the trace

$$Z = \operatorname{tr} \Lambda^n. \tag{1.24}$$

As it is easy to see, for a lattice which is infinite in one direction, the largest

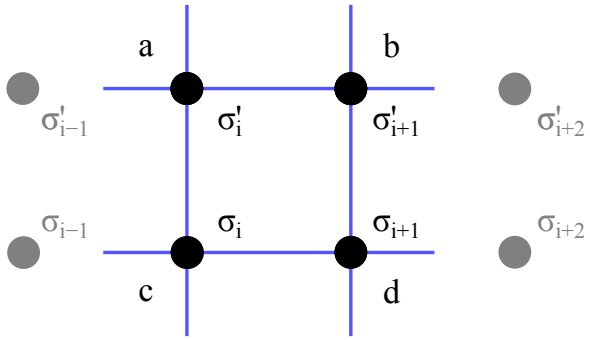

Figure 1.3: Standard spin labels arrangement for weight $w(a, b, c, d)$

eigenvalue of Λ corresponds to the value of the partition function per row, and therefore the partition function per site is

$$\kappa = \lim_{m,n\to\infty} (\Lambda^n)^{\frac{1}{mn}} = \lim_{m\to\infty} \lambda^{\frac{1}{m}} \tag{1.25}$$

where λ is the largest eigenvalue of Λ. If we keep m fixed, we then obtain the partition function of the lattice, infinite only in the n direction, or we could take m to infinity and get the infinite-lattice partition function. For numerical calculations one could use sufficiently large m and then extrapolate to $m \to \infty$.

There are several problems arising in that approach. First of all, the size of Λ grows exponentially with the size of the lattice. On top of this, diagonalisation algorithms usually scale like $O(N^3)$, so there is a quite hard limit for the size of the lattice, it cannot be larger than a dozen sites or so. Moreover, if we add more sites, improvements are not as good as desired, because a part of the spectrum which is close to the largest eigenvalue depends on the size of the system. In most cases the degree of degeneracy of the level next to the largest eigenvalue is proportional to the number of sites, and the gap between the level and the largest eigenvalue is proportional to the distance (in parameter space) to the critical point. As one can see, the Transfer Matrix method is not very appropriate for calculations near the critical point, and thus checking the scaling hypothesis with it would be extremely difficult.

Many other methods exist, ranging from the very specific and semi-analytic to the rather general. We consider only one, but it is both a general and powerful method which works reasonably well near the critical point. It was first proposed by Baxter [34] in his early work on a dimer problem. The method was shown to

be very efficient for computer implementation even at that time. Later Baxter used so called the Corner Transfer Matrices to present an alternative solution to the eight-vertex model [27] and various other problems. There has been a lot of follow-up work (see [35] and references therein) where the method is successfully used to tackle various statistical mechanics problems.

We describe the method following Baxter's two main papers on the numerical CTM [34, 36]. We don't touch on any methods to solve the final equations in this chapter, since the entire Chapter 2 is dedicated to the solution methods and their physical meaning.

The basic idea is that λ can be found from the variational expression

$$\lambda = \max_{V \neq 0} \frac{V^\dagger \Lambda V}{V^\dagger V}. \tag{1.26}$$

Let's construct a trial vector V for the eigenvector of the transfer matrix Λ, which corresponds to its largest eigenvalue λ.

$$V(\sigma_1, \sigma_2, \sigma_3 \ldots, \sigma_m) = \ \mathrm{tr}\,[F(\sigma_1, \sigma_2)F(\sigma_2, \sigma_3) \cdots F(\sigma_m, \sigma_1)], \tag{1.27}$$

where $F(a, b)$ are given by N by N matrices with the following symmetry:

$$F = \begin{pmatrix} F(1,1) & F(1,2) \\ F(2,1) & F(2,2) \end{pmatrix} = F^\dagger. \tag{1.28}$$

Now we use (1.27) to find an extremum of (1.26). Consider a denominator $V^\dagger V$ first and use a trial eigenvector V to compute it:

$$V^\dagger V = \ \mathrm{tr}\, R^m, \tag{1.29}$$

where the matrix R is similar to a column-to-column transfer matrix, but is constructed from two F matrices:

$$R(\lambda, a, \lambda' | \mu, b, \mu') = F_{\lambda\mu}(a, b) F_{\lambda', \mu'}(a, b). \tag{1.30}$$

Here R is a $2N^2 \times 2N^2$ matrix, λ, λ', μ and μ' run from 1 to N. Let X be the eigenvector of R corresponding to the largest ξ. For large m we have

$$V^\dagger V \sim \xi^m. \tag{1.31}$$

Since X is an eigenvector, it follows that

$$\sum_b F(a,b)X(b)F(b,a) = \xi X(a). \tag{1.32}$$

From the definition of R it follows that R is a symmetric matrix, hence its largest eigenvalue ξ can be found using a variational principle:

$$\xi = \frac{\sum\limits_{a,b} \mathrm{tr}\,[X^\dagger(a)F(a,b)X(b)F(b,a)]}{\sum\limits_a \mathrm{tr}\, X^\dagger(a)X(a)}. \tag{1.33}$$

Now consider the numerator of expression (1.26) in the same way. Using (1.27) one can write

$$V^\dagger \Lambda V = \mathrm{tr}\, S^m, \tag{1.34}$$

where S is a matrix with matrix elements

$$S(\lambda, a, a', \lambda' | \mu, b, b', \mu') = F_{\lambda\mu}(a,b)w(a,b,a',b')F_{\lambda'\mu'}(a',b'). \tag{1.35}$$

Using the same arguments as for $V^\dagger V$ we have

$$V^\dagger \Lambda V \sim \eta^m, \tag{1.36}$$

where η is the largest eigenvalue of S. Let the corresponding eigenvector be Y, then the eigenvalue equation is

$$\sum_{b,b'} w(a,b,a',b')F(a,b)Y(b,b')F(b',a') = \eta Y(a,a'), \qquad a,a' = +,-. \tag{1.37}$$

Again, since S is a symmetric matrix, it is possible to find η using the variational expression

$$\eta = \frac{\sum\limits_{a,b,a',b'} w(a,b,a',b')\ \mathrm{tr}\,[Y^\dagger(a,a')F(a,b)Y(b,b')F(b',a')]}{\sum\limits_{a,a'} \mathrm{tr}\, Y^\dagger(a,a')Y(a,a')}. \tag{1.38}$$

Substituting η and ζ back into equation (1.26) yields

$$\kappa = \lambda^{1/m} = \frac{\eta}{\xi}. \tag{1.39}$$

To find vectors X and Y one could differentiate (1.32) and (1.37) with respect to the Ansatz matrix F, keeping in mind that the solution of these equations is stationary in respect of X and Y. This gives an additional relation between X and Y:

$$\sum_{a',b'} w(a,b,a',b')Y(b,b')F(b',a')Y(a',a) = \eta' X(b)F(b,a)X(a), \tag{1.40}$$

where

$$\eta' = \kappa \frac{\sum\limits_{a,a'} \operatorname{tr} Y(a',a)Y(a,a')}{\sum\limits_{a} \operatorname{tr} X^2(a)}. \tag{1.41}$$

Together with equations (1.32) and (1.37) this equation defines matrices F, X and Y up to some normalisation factors, making it possible to compute the partition function per site κ.

1.2.1 Main Equations

The usual row-to-row Transfer Matrix approach has a feature that can sometimes be unsatisfactory, namely the absence of explicit symmetry between rows and columns.

Let us make the following substitution for the extremal vectors X and Y:

$$X(a) = A^\dagger(a)A(a), \tag{1.42}$$

$$Y(a,b) = A^\dagger(a)G(a,b)A(b), \tag{1.43}$$

with a symmetry condition for G:

$$G(a,b) = G^\dagger(b,a) \tag{1.44}$$

(essentially the same condition as for F).

Now we can write equations that determine A, F and G in the following form:

$$\sum_{b,b'} w(a,b,a',b')F(a,b)A^\dagger(b)G(b,b')A(b')F(b',a') = \quad \eta A^\dagger(a)G(a,a')A(a'), \tag{1.45a}$$

$$\sum_{b} F(a,b)A^\dagger(b)A(b)F(b,a) = \quad \xi A^\dagger(a)A(a), \tag{1.45b}$$

$$\sum_{b,b'} w(a',a,b',b)G(a,b)A(b)F(b,b')A^\dagger(b')G(b',a') = \quad \eta' A(a)F(a,a')A^\dagger(a'), \tag{1.45c}$$

$$\sum_{b} G(a,b)A(b)A^\dagger(b)G(b,a) = \quad \xi' A^\dagger(a)A(a). \tag{1.45d}$$

These equations are called the Corner Transfer Matrix equations, and the matrix $A(a)$ is called the Corner Transfer Matrix. Matrices F and G are called the Half-Row and Half-Column Transfer Matrices respectively. The basis can be changed to make $A(a)$ diagonal, as will be considered in detail in Chapter 2.

There is a natural graphical representation of (1.45) that is shown in Figures 1.4, 1.5.

Naively one could construct such equations just by considering a corner of the lattice and attaching a weight and two stripes to it. But as was observed with one of the algorithm variants for the honeycomb lattice, this scheme might not work very well. Possible explanations include the connection of the algorithm to White's Renormalisation Group, which we'll consider in Chapter 2.

Recalling an expression for the partition function per site one could write the partition function via A, F and G. Since the expression for κ is stationary with respect to variations of A, F and G, derivatives of κ could be computed directly. The only price to pay is somewhat lower precision for the derivatives, i.e., if the error in A, F and G is proportional to ϵ, than the error in κ is about ϵ^2 and for the derivatives the error is again of order ϵ.

From the solution of the main equations the observables can be readily determined. Since the partition function per site equals η/ξ, it can be expressed as

$$\kappa = \frac{r_1 r_4}{r_2 r_3} \tag{1.46}$$

where r_i are different lattice configurations in which every scalar factor in A, F

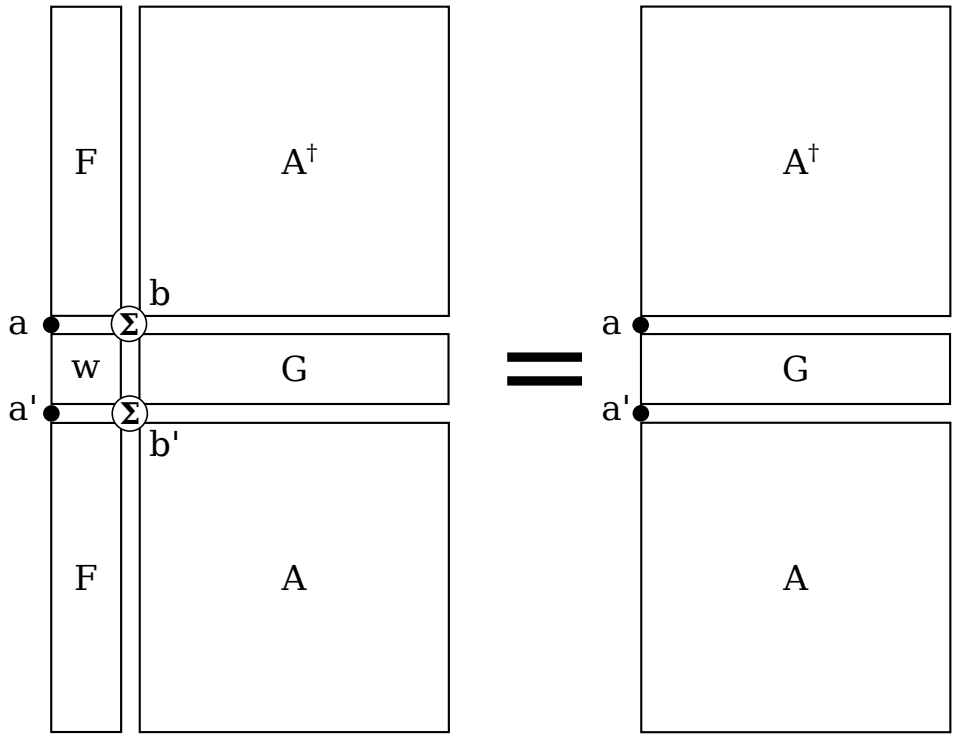

Figure 1.4: Graphical representation of the relations (1.45a) and (1.45c).

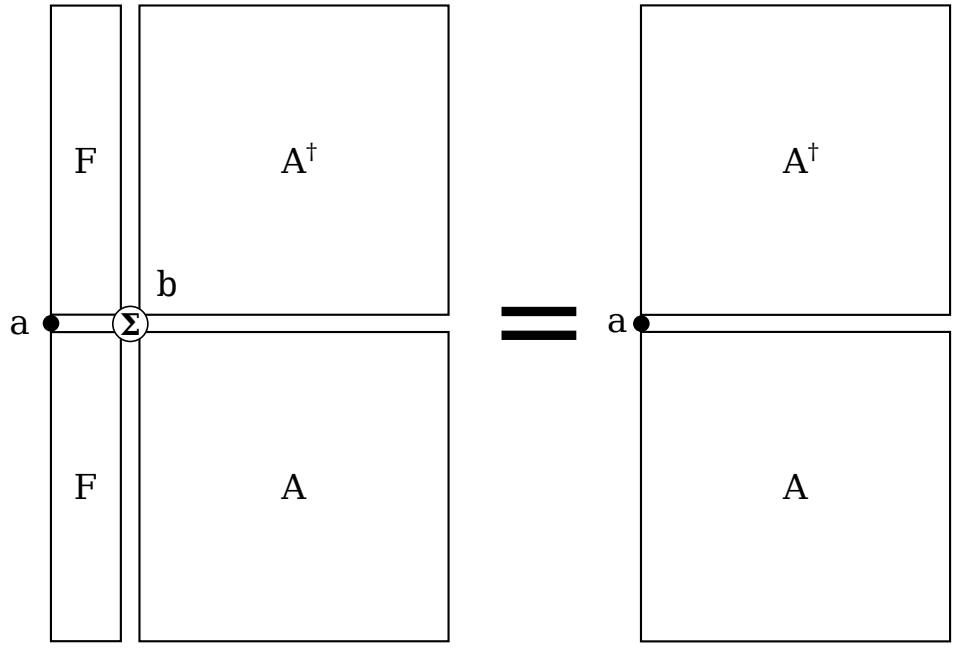

Figure 1.5: Graphical representation of the relations (1.45b) and (1.45d).

and G would cancel, but w stays:

$$r_1 = \sum_a \operatorname{tr} A(a)A^\dagger(A)A(a)A^\dagger(a), \tag{1.47a}$$

$$r_2 = \sum_{a,b} \operatorname{tr} A(a)F(a,b)A^\dagger(a)A(b)F(b,a)A^\dagger(a), \tag{1.47b}$$

$$r_3 = \sum_{a,b} \operatorname{tr} A^\dagger(a)G(a,b)A(a)A^\dagger(b)G(b,a)A(a), \tag{1.47c}$$

$$r_4 = \sum_{a,a',b,b'} w(a,b,a',b') \operatorname{tr} A^\dagger(a')G(a',a)A(a)F(a,b)A^\dagger(b)G(b,b')A(b')F(b',a'). \tag{1.47d}$$

For the Ising Model there are two parameters in w: the external field H and the temperature β. The logarithmic derivative with respect to H gives the average value of the spin (magnetisation) as

$$M = \langle \sigma_i \rangle = \frac{\partial \log \kappa}{\partial H} = r_1^{-1} \sum_a a \operatorname{tr} \Big[A(a)A^\dagger(a)A(a)A^\dagger(a) \Big]. \tag{1.48}$$

The internal energy is given by the logarithmic derivative with respect to the inverse temperature β:

$$U = \langle \sigma_i \sigma_{i+1} \rangle = \frac{\partial \log \kappa}{\partial \beta} = r_2^{-1} \sum_{a,b} ab \cdot \operatorname{tr} [A(a)F(a,b)A^\dagger(b)A(b)F(b,a)A^\dagger(a)]. \tag{1.49}$$

As already mentioned, there are several definitions of the internal energy. In [36] and many other papers, the internal energy is defined in a way similar to $U' = -\langle \log w \rangle$ or in full form for the square lattice as

$$\begin{aligned} U' &= -T^2 \tfrac{\partial \log Z(T,H)}{\partial T} = -2\beta \langle \sigma_i \sigma_{i+1} \rangle - H \langle \sigma_i \rangle = -r_4^{-1} \sum_{a,b,a',b'} \log w(a,b,a',b') \\ &\quad \times \operatorname{tr} \Big[A^\dagger(a')G(a',a)A(a)F(a,b)A^\dagger(b)G(b,b')A(b')F(b',a') \Big], \end{aligned} \tag{1.50}$$

where $Z(T,H)$ is defined as in (1.3) with $\beta = J/T$. This definition makes sense especially when the $1/T$ factor is included in the magnetic term in the Hamiltonian.

Both equations (1.49) and (1.50) are correct, and simply related. Equation (1.50) represents the bond energy per face with magnetic energy $M(H) \cdot H$ corresponding to the spins of the lattice. Since there are a different number of spins and bonds per face for different kinds of lattice, Table 1.1 should be used to relate

U and U' for the other types of lattice.

The expression (1.49) can be generalised to compute not just the correlation between adjacent sites, but sites on the same cardinal line and arbitrary distance apart, with some technical limitations (see formula (2.16)). In general, the matrix R is quite large for practical computations, even if we want only its largest eigenvalue ξ to compute the correlation length, although it is possible to compute the correlation length for short distances directly.

Chapter 2

Corner Transfer Matrix Algorithm

The solution of the system of matrix equations (1.45) by general methods is rather difficult. While the original publication [36] proposed an iterative method to solve the system, it was prone to oscillations and using a general method (Newton-Raphson procedure) was suggested for such cases. As it turned out later, a better method exists, which makes much more physical sense than the general methods. In this chapter we describe the method in detail along with its variants and generalisations.

2.1 Renormalisation Group

The concept of Renormalisation Group (RG)[1] is related to the Scaling Hypothesis. It was first thoroughly used in the framework of Quantum Field Theory, where it takes the form of Gell-Mann-Low RG and Callan-Symanzik equation for the beta-function which shows how an effective coupling "constant" changes with the change of energy scale. In addition RG was employed to perform a renormalisation programme, helping with the not entirely rigorous tricks of cancelling divergent values in various QFT models.

The second wave of RG-related methods came with the work of Fisher [10], Kadanoff [37], Polyakov [38] and others. Their basic idea was that many measurable quantities in condensed matter physics, in particular in statistical mechanics,

[1] It was noted in many places that the name "Renormalisation Group" is a misnomer, since the actual transformations usually form a semigroup, and not a group in the strict mathematical sense.

are covariant to some scaling transformation. They used this property as a general tool for deriving various relations mostly for the correlation functions of the various statistical mechanics models without any relation to the underlying lattice. Also they defined very important properties of various physical values, namely critical indices, which act like dimension in the scaling transformation. In addition to various results in lattice statistical mechanics and condensed matter theory, RG applications include one of the most prominent theories in mathematical physics – Conformal Field Theory. Some of these aspects of scaling will be considered in Chapter 3, since they are crucial for obtaining the scaling function.

In the context of direct application to lattice statistical mechanics models, the RG is mostly known due to the work of Wilson [39], who very successfully applied a form of RG to the Kondo problem [40]. Wilson proposed a robust method for computing the free energy of various systems numerically. The method is intuitively quite simple: we take a small part of the system (a block), compute its spectrum by some Hamiltonian diagonalisation algorithm, then take the part of spectrum that we are mostly interested in (for statistical mechanics systems this is usually the lowest-lying levels), and build an effective Hamiltonian of the large system joining two (or more) blocks together. The solution of the original Hamiltonian is a fixed point of such a transformation, i.e., the point where the spectrum of the effective Hamiltonian does not change any more by RG transformation. Numerically, fixed points are usually obtained by some form of iteration, which are considered as elements of the group. One of the key points of the method is a choice of the levels which will constitute the effective Hamiltonian at the next iteration step. Wilson chose the lowest lying levels, which seemed to be rather obvious for the lattice models in equilibrium. While the method is very simple and can produce excellent results[2], it turned out that for systems with strong coupling and long-range correlations the method may not yield good results, and mostly impurity-related problems, such as the Kondo problem, could be solved with satisfactory precision. There are better choices for the levels selection.

2.1.1 Density Matrix Renormalisation Group

One such choice comes from consideration of the block boundaries, and observation of the incorrect treatment of these boundaries [41]. The key changes of the method is a choice as to what part of the spectrum to use for the effective

[2]The method was one of the main reasons for awarding Wilson a Nobel prize relatively quickly after publication.

Hamiltonian. Instead of choosing the lowest levels of the effective Hamiltonian, we compute a density matrix (DM) of part of the Hamiltonian (e.g., the DM of one of the blocks from the last step) and choose the levels with the highest weights in the DM [14, 42]. This fixes the problem with boundaries and allows for much more precise computations with the same size of effective Hamiltonian representation.

The basic algorithm follows; we outline only the version for the infinite system (without boundaries), many other versions have appeared since the original publication. The version for the finite systems is slightly more complicated – the iterations "sweep" the system back and forth. While the original algorithm is somewhat more detailed, here we present a hopefully correct interpretation, that is more abstract and not yet ready for immediate implementation, but on the other hand it is more understandable in terms of the general idea.

1. Take a small block of the system, small enough to diagonalise its Hamiltonian. One side of the block has a free boundary condition, e.g., on the left.

2. Add a very small block (e.g., one site) to the right side of the main block from the previous step.

3. Construct a "full" system by mirroring the extended block, and then diagonalise its Hamiltonian.

4. Compute a Density Matrix of the left part of the system in respect to the "outer" system (the right part), using the eigenstates of the Hamiltonian from the previous steps.

5. Diagonalise the Density Matrix and take the largest eigenvalues, leaving out those which contribute the least to the state of the whole system.

6. Construct an effective Hamiltonian from the truncated Density Matrix, keeping everything in the basis where the Density Matrix is diagonal. The effective Hamiltonian is now an input Hamiltonian for the next iteration.

7. Go to Step 2

While the algorithm intrinsically uses a quantum mechanics concept – the density matrix, it turns out that it can be applied (after proper adaptation) to classical systems. There is a strong connection between DMRG and variational

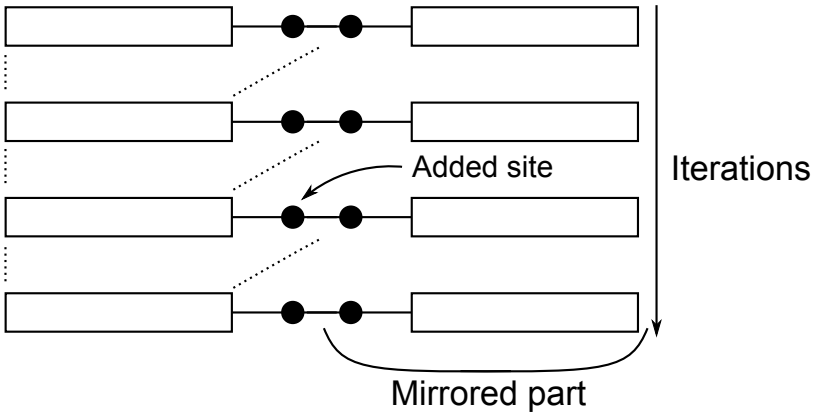

Figure 2.1: White's DMRG algorithm iteration scheme.

CTM approaches. The main point of the connection is that the spectrum of the CTM corresponds to the spectrum of the Density Matrix. Hence by doing the iterations in such a way that all objects are represented in the basis which keeps only the largest eigenvalues of the Density Matrix (or CTM) we work with the best possible efficiency for the given sizes of all matrices.

There is a plethora of algorithms stemming from the original DMRG. A good review of them is [43]. The Corner Transfer Matrix Renormalisation Group is among them, and it is the main algorithm that we will use.

2.1.2 Corner Transfer Matrix Renormalisation Group

In the original work Baxter proposed an iterative scheme to solve the non-linear matrix equations (1.45). While in some cases such a scheme can be efficient and convergent, in the most interesting region of the parameter space – the vicinity of the critical point – the scheme does not converge, oscillating wildly around the fixed point. One could compare this behaviour with the oscillation of the usual simple Newton procedure when solving a non-linear equation.

In [36] the failure of the proposed algorithm was anticipated. The Newton-Raphson algorithm was proposed as a replacement for the critical region, or wherever the main iteration scheme fails to converge. While sometimes using this algorithm is inevitable, it must be a "last resort", since it might be difficult to achieve the same properties of the CTM as in the real RG-method, i.e., its intrinsic correspondence to the density matrix of the related 1D quantum model.

There is a profound correspondence between d-dimensional quantum systems and $(d+1)$-dimensional statistical mechanics systems. Moreover, there is a cor-

respondence between time and temperature. It won't be described in detail here, some aspects of that correspondence are still very poorly understood, but a mapping between different quantities in 1D-quantum mechanics and 2D-statistical mechanics can be established.

The basic idea for applying DMRG to the CTM algorithm is that the Density Matrix corresponds to the CTM at least in some, not very rigorous sense. Detailed argumentation for this argument can be found in the original papers by Nishino [15, 16]. Here we outline only the classical algorithm without any links to its quantum roots.

Let's define a simple symmetrical weight for the square lattice Ising model

$$w(a,b,c,d) = \exp\left[\frac{\beta}{2}(ab+ac+bd+cd) + \frac{H}{4}(a+b+c+d)\right], \tag{2.1}$$

so the CTM and Half-Row Transfer Matrix (HRTM) corresponding to this weight are always symmetrical and no special consideration when diagonalising the CTM is required.

The steps of the CTMRG algorithm now follow:

1. Construct an initial CTM $A(a)$ and HRTM $F(a,b)$. For some systems it matters whether the initial state is close enough to the solution, but for the Ising model any state would work.

2. Extend $A(a)$ by attaching two $F(a,b)$ and a weight w to form an extended CTM, which is (for systems with two spin states) twice as large as the original $A(a)$.

$$H'(c;a,d) = \sum_b w(a,b,c,d)F^\dagger(b,a)A(b)F(b,d). \tag{2.2}$$

   Note that while the F should be symmetric, so $\dagger$ and transposition of a and b play no role; it is better to have them in place. One reason is to damp numerical fluctuations in F. Another reason is that with such modifications the algorithm could be readily employed with the weights (2.19) (or (2.20)) for the triangular lattice.

3. Diagonalise $H(a)$ by computing its eigenvectors and eigenvalues. This is

numerically the most expensive step of the algorithm.

$$\begin{aligned} H(a) &= \begin{pmatrix} H'(a;1,1) & H'(a;1,2) \\ H'(a;2,1) & H'(a;2,2) \end{pmatrix} \\ &= \begin{pmatrix} U_{11}(a) & U_{12}(a) \\ U_{21}(a) & U_{22}(a) \end{pmatrix} \begin{pmatrix} \mathcal{A}_1(a) & \\ & \mathcal{A}_2(a) \end{pmatrix} \begin{pmatrix} U_{11}(a) & U_{11}(a) \\ U_{21}(a) & U_{22}(a) \end{pmatrix}^{\dagger} \end{aligned} \quad (2.3)$$

We assume that eigenvalues of $\mathcal{A}(\alpha)$ are arranged in decreasing order. Sizes of each $\mathcal{A}_i(a)$ and $U_{ij}(a)$ are $N \times N$.

4. Take the most significant part of the $A(a)$ spectrum and the corresponding eigenvectors. For two-state systems this usually means that we take the largest half of all eigenvalues (if we want to recur to the beginning of the algorithm)

$$\tilde{A}(a) = \mathcal{A}_1(a). \quad (2.4)$$

5. Build a new extended HRTM from the $F(a,b)$ and weight w

$$F'(a,b;c,d) = w(a,b,c,d) \cdot F(b,d). \quad (2.5)$$

6. Use the eigenvectors from Step 3 to convert (and "project") the extended HRTM to the new basis.

$$\tilde{F}(a,b) = \left(U_{11}^{\dagger}(a) U_{21}^{\dagger}(a)\right) \begin{pmatrix} F(a,1;b,1) & F(a,1;b,2) \\ F(a,2;b,1) & F(a,2;b,2) \end{pmatrix} \begin{pmatrix} U_{11}(b) \\ U_{21}(b) \end{pmatrix}. \quad (2.6)$$

7. Since now we have both A and F with the sizes identical to the beginning of the iteration, we can start the next iteration from Step 2.

Some graphical depictions of these steps are given in Figures 2.2 and 2.3.

In contrast with Baxter's iteration scheme [36], for the Ising model this algorithm always converges, albeit with convergence speed very close to the critical point obviously slower than for any other values of H and β.

During evaluation of the algorithm several useful observations were made. Most of them are quite trivial, but nevertheless worth noting. The first observation concerns convergence of the iteration while changing parameters of w, e.g.,

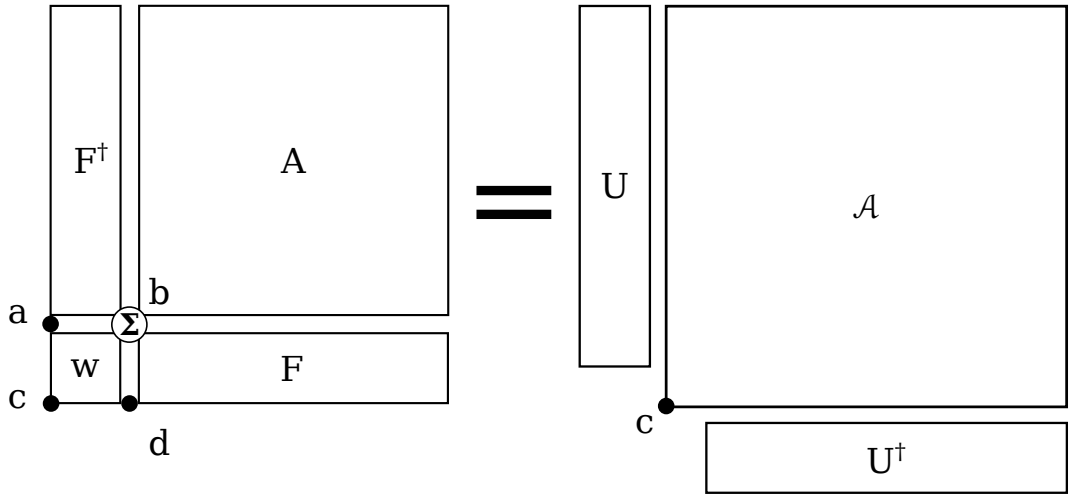

Figure 2.2: Extending and diagonalising the CTM, formulas (2.2) and (2.3).

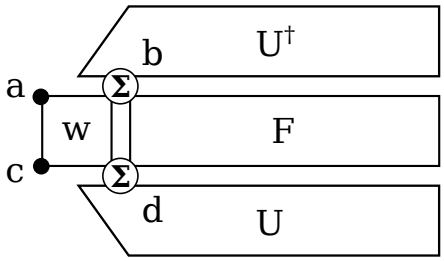

Figure 2.3: HRTM extending and basis change step, formulas (2.5) and (2.6).

the temperature. It was found that the algorithm converges faster when the temperature was increased rather decreased, in other words, it is easier to heat up the system than to cool it down. A similar effect should take place while changing the strength of the magnetic field, but no actual observations were made in this case.

The heating-cooling iteration speed difference might play some role in practical computation, but only if parameters are changed when N is large enough. In practice this effect is barely used, because it is clear that for the calculations that are required to obtain the scaling function it is not necessary to change any parameters while N is large, since one could change β and H for smallest N and then only change (increase) N.

One of the most noticeable differences between the idealised equations and real calculations is that the HRTM $F(a, b)$ could in fact fail to have the symmetry of the original Ansatz (1.28). This symmetry is required for the matrices S (1.35) and R (1.30) to be symmetric, so that the variational method can be used to find their eigenvalues. As it turned out, non-symmetry of F plays no role in

algorithm performance, that is, performance doesn't differ between triangular and square lattices (for the latter $F = G^\dagger$, but F itself is non-symmetric, see Section 2.2.2 below). The fact that for the triangular and honeycomb lattices the scaling function is totally identical to the scaling function for the square lattice indicates that F is not required to be symmetric.

2.2 Variations of the Algorithm

2.2.1 Square Lattice

Baxter's original algorithm for solving (1.45) was devised for a slightly more general Boltzmann weight

$$\begin{aligned} w(a,b,c,d) &= \exp\Big[C + \frac{\beta}{2}(ab+ac+bd+cd) + \frac{H}{4}(a+b+c+d) \\ &+ J_d(ad+bc) + J_3(abd+abc+acd+cdb) + J_4 abcd\Big], \end{aligned} \quad (2.7)$$

which obviously contains the Boltzmann weight of the square lattice Ising model.

While the CTM can be still made diagonal in this case, the HRTMs for rows and columns are different. The extended CTM from (2.2) is constructed as

$$H'(c;a,d) = \sum_b w(a,b,c,d)G(a,b)A(b)F(b,d), \quad (2.8)$$

and then Singular Value Decomposition [44] is used, which allows us to keep $A(a)$ diagonal, while having different but symmetric HRTMs for rows and columns. Any effects of having G transposed in (2.8) in the same way as the first F in (2.2) were not studied.

The SVD decomposition of $H'(c)$ is

$$\begin{aligned} H'(a) &= \begin{pmatrix} H'(a;1,1) & H'(a;1,2) \\ H'(a;2,1) & H'(a;2,2) \end{pmatrix} \\ &= \begin{pmatrix} U_{11}(a) & U_{12}(a) \\ U_{21}(a) & U_{22}(a) \end{pmatrix} \begin{pmatrix} \mathcal{A}_1(a) & \\ & \mathcal{A}_2(a) \end{pmatrix} \begin{pmatrix} V_{11}(a) & V_{11}(a) \\ V_{21}(a) & V_{22}(a) \end{pmatrix}^\dagger \end{aligned} \quad (2.9)$$

The HRTM extension and projection step have a form similar to (2.5):

$$F'(a,b;c,d) = w(a,b,c,d) \cdot F(b,d), \tag{2.10a}$$
$$G'(c,a;d,b) = w(a,b,c,d) \cdot G(a,b), \tag{2.10b}$$

and to (2.6):

$$\tilde{F}(a,b) = \left(V_{11}^{\dagger}(a)V_{21}^{\dagger}(a)\right) \begin{pmatrix} F(a,1;b,1) & F(a,1;b,2) \\ F(a,2;b,1) & F(a,2;b,2) \end{pmatrix} \begin{pmatrix} V_{11}(b) \\ V_{21}(b) \end{pmatrix}, \tag{2.11a}$$

$$\tilde{G}(a,b) = \left(U_{11}^{\dagger}(a)U_{21}^{\dagger}(a)\right) \begin{pmatrix} G(a,1;b,1) & G(a,1;b,2) \\ G(a,2;b,1) & G(a,2;b,2) \end{pmatrix} \begin{pmatrix} U_{11}(b) \\ U_{21}(b) \end{pmatrix}. \tag{2.11b}$$

All other details of the algorithm are the same.

The real requirement for the weight is not be in the form (2.7), but to be symmetrical to reflections:

$$w(c,d,a,b) = w(a,b,c,d) = w(b,a,d,c). \tag{2.12}$$

This condition is required for the algorithm to reconstruct the whole lattice by reflections from the single CTM which could be non-symmetrical as in (2.8). If this condition doesn't hold, then additional work on the algorithm is required to make it work. It sometimes happens that the required symmetry is much less than it seems at first, but at least a loose relation to the density matrix should be maintained.

Edge and spin sharing between adjacent weights. In all expressions for Boltzmann weights that we have already presented it should be noted that edges and spins are shared between adjacent weights in construction of the lattice. This is especially important when one does various lattice transformations, since the natural "sharing" factors for the spins could differ for the lattices that could be transformed into each other.

The following weight would give identical results (if the edge cases are worked out):

$$w(a,b,c,d) = \exp\left[\beta((xab + (1-x)cd) + (yac + (1-y)bd))\right] \tag{2.13}$$

for any x and y between 0 and 1. Usually the edges are split evenly, so $x = y = \frac{1}{2}$. "Shares" of the spins could also be distributed unevenly. Uneven distribution of

spin shares in the weight gives an interesting possibility of simulating the lattice with one spin "marked", i.e., having a different field strength from all other spins. Consider the following weight

$$w(a,b,c,d) = \exp\left[\frac{\beta}{2}(ab+cd+ac+bd) + \frac{H}{2}[x(a+d)+(1-x)(b+c)]\right]. \tag{2.14}$$

If $x \neq \frac{1}{2}$, it is nonetheless clear that if we draw the lattice made of such "rhombic" spins, it would be equivalent to the usual square lattice, with one difference. Since we construct a CTM for one quarter of the lattice only, the central spin would be multiplied not by H, but by $2(1-x)H$, as can be clearly seen from the formula for the magnetisation (1.48). Other formulas, namely for the free energy F and internal energy U require more complex modifications.

It also turns out that modification of the weight is not required in order to alter the central spin field. One could simply add a correcting factor with the required field strength to (1.48) without changing the weight and without recomputing $A(a)$:

$$A'(a) = A(a)\exp\left[\frac{1}{4}(H'-H)a\right]. \tag{2.15}$$

As with the weight changing method, formulas for other observables would require similar modifications to make them consistent.

This possibility of freely changing the field at least at one of the lattice sites is quite interesting, but it has not been explored here.

Computing Correlation Functions. The formula (1.49) allows us to compute a correlation of the adjacent (in cardinal direction) spins, thus giving a value for the internal energy. It is possible to compute the correlation function not just for adjacent, but for spins separated by an arbitrary number of bonds. Simply by adding an arbitrary number of F-matrices we obtain

$$\langle \sigma_i \sigma_{i+n} \rangle = \frac{r_{11} + r_{22} - r_{21} - r_{12}}{r_{11} + r_{22} + r_{21} + r_{12}} \tag{2.16}$$

where

$$\begin{aligned} r_{ab} &= \sum_{k_i, i=1..n-1} ab \ \mathrm{tr}\Big[A(a)F(a,k_1)F(k_1,k_2)F(k_2,k_3)\cdots F(k_n,b)A(a) \\ &\times \ A(a)F^\dagger(b,k_n)\cdots F^\dagger(k_3,k_2)F^\dagger(k_2,k_1)F^\dagger(k_1,a)\Big], \end{aligned} \tag{2.17}$$

see Figure 2.4. We tested the formula in the critical point for the square lattice for $n = 1 \ldots 5$, with the exact values provided in [45]. The size of the CTMs was 64, and for all n precision better than 10^{-6} was easily achievable if enough CTMRG iterations were made.

While it is clear that the approach worked well for spins in the same row or columns, it is not yet clear how to generalise it to arbitrary spin locations while keeping it efficient. Large n also presents a problem, with the most obvious solution being to diagonalise the "transfer matrix"

$$F(a, \alpha, b|a', \alpha', b') = F_{aa'}(\alpha, \alpha')F_{b'b}(\alpha, \alpha'), \tag{2.18}$$

but its size is $2N^2 \times 2N^2$, so for a large CTM the size of that matrix would be too big for a full diagonalisation.

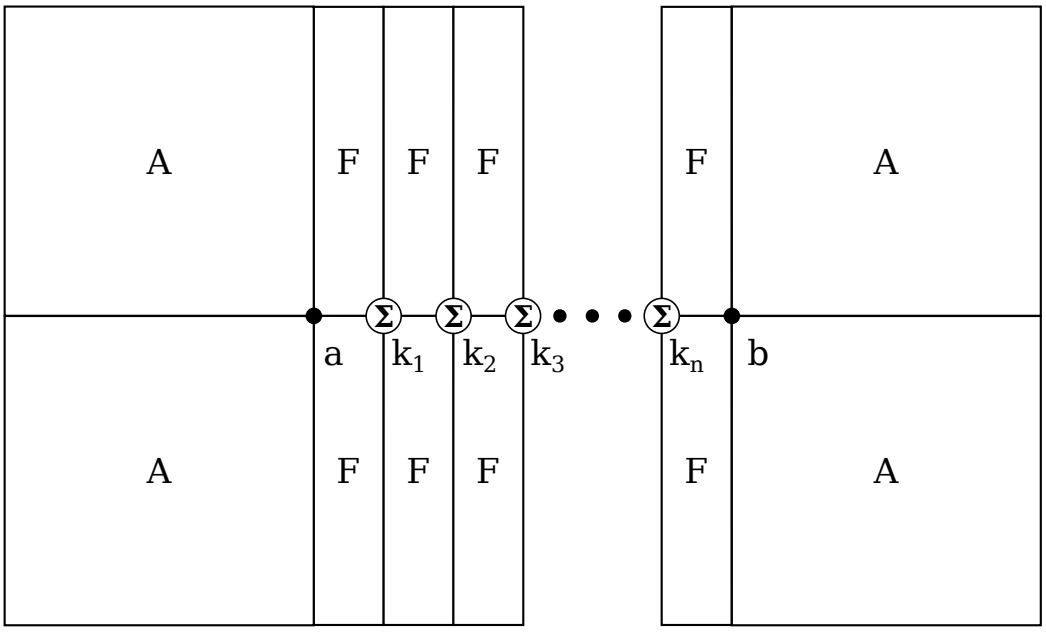

Figure 2.4: Computing the correlation function.

2.2.2 Triangular lattice

In the work of Tsang [46], Baxter [47, 48] and Malakis [49, 50] the 2D Ising Model on the triangular lattice was treated specially, with a separate algorithm, which would require a rewrite of the computer program. It turned out that such an approach is not necessary, since it is possible to use a special weight that

represents a triangular lattice inscribed into the square one, with weight

$$w(a,b,c,d) = \exp\left[\frac{\beta}{2}(ab+ac+bd+cd+2ad)+\frac{H}{4}(a+b+c+d)\right], \quad (2.19)$$

and a modification (2.15) to the observable value functions, i.e., if we just take A^6, then the field strength in the centre of the lattice will be $\frac{3}{2}H$ instead of H. This weight was used for all high-precision calculations for the triangular lattice.

There is also an equivalent weight

$$w(a,b,c,d) = \exp\left[\frac{\beta}{2}(ab+ac+bd+cd+2bc)+\frac{H}{4}(a+b+c+d)\right] \quad (2.20)$$

which was shown to give totally identical results to the weight (2.19). Of course, for the weight (2.20) we compute A^3 for the magnetisation value and adjust all other related formulas accordingly.

It is possible to write down a weight that does not require any significant modifications for computing the observable functions. The only change will be different factors for spins, that will be proportional to the angle that the particular weight shares in the spin, so if we draw a lattice then it will look like a rhombic lattice, see Figure 2.5. The third weight in that picture doesn't require any special correction factors (similar to (2.15)) in the observables formulas. It has spins sharing factors proportional to the angles adjacent to the spin in the real triangular lattice, with

$$w(a,b,c,d) = \exp\left[\frac{\beta}{2}(ab+ac+bd+cd+2ad)+\frac{H}{6}(2a+b+c+2d)\right]. \quad (2.21)$$

This weight was eventually tested, including a thorough comparison to the results of weight (2.19), but was not used in real calculations because no real difference was found.

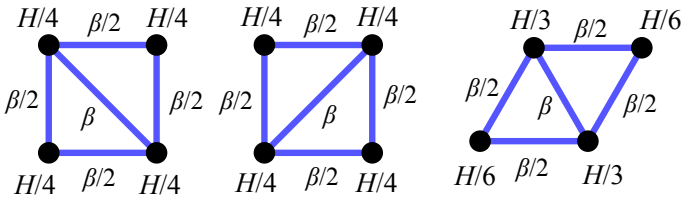

Figure 2.5: Variants of the triangular weights for the square-lattice algorithm.

It should be noted that some symmetries used for the derivation of CTM equations in Chapter 1 don't hold any longer in the case of the triangular lattice. Symmetry of F no longer holds, and the actual computed F is indeed non-symmetrical, but this makes no difference to the CTMRG. Even SVD modification of the algorithm is not needed, the reason is that an extended CTM is a symmetrical matrix in this case.

2.2.3 Honeycomb lattice

There is no transformation of the honeycomb lattice to the square one as simple as for the triangular lattice. Three algorithms were proposed and tested, and while two algorithms performed quite well, the first gave bad results which prevented it from actual usage. While no complete investigation of a reason for failure was done (aside from excluding programming errors and checking internal consistency as thoroughly as possible), it was suggested that the inability of the algorithm to be mapped back to DMRG might contribute to its failure. We present all three algorithms with emphasis on the third one which was actually used for most of the practical honeycomb lattice computations.

"Auxiliary Matrix" algorithm. In this algorithm we use the standard "brick-wall" transformation[3] to inscribe a honeycomb lattice into a rectangular one. The Boltzmann weight takes the form

$$w(a,b,c,d) = \sum_i \exp\left[\beta(ai + ci + di) + \frac{H}{6}(2a + b + c + 2d + 6i)\right]. \qquad (2.22)$$

The first difference from the simple square and triangular lattices is that there is an "internal" spin which we are summing over. While that was proposed as the reason for the aforementioned problems, it turned out that algorithms with the square lattice built from weights with internal summation are performing rather well, so summation is not a cause of any problems. Also because of the problems, the algorithm is not discussed in full.

To test if summation is a root of the problems, we used the square lattice Ising model with some alternative weights. The first comparison weight to try was

$$w(a,b,c,d) = \exp\left[\beta(ad + bc) + \frac{H}{4}(a + b + c + d)\right], \qquad (2.23)$$

[3]It is depicted on the cover of the original publication of Baxter's book [20].

and although it doesn't have any summation of internal spin states, it has another interesting property. This weight actually represents two disconnected square lattices, and if we attempt to calculate the correlation between two adjacent spins (in the cardinal direction), then it will be zero for any temperature. The zero correlation was actually verified. The weight is also used for the analytical solution in [27]. The reason it was used for the internal spin summation test is that it is similar to the weight with summation

$$w(a,b,c,d) = \sum_i \exp\left[\beta(ai+bi+ci+di) + \frac{H}{4}(a+b+c+d+4i)\right], \quad (2.24)$$

which is very similar to (2.22). The weight (2.24) represents a single lattice and should have all properties equal to the usual square lattice weight (2.1).

The procedure for computing A and F for the test weights (2.23) and (2.24) is the same as for the usual square weight since the symmetry properties of the weights are the same. No significant deviations between all three weights were found, although the convergence speed for the weight (2.23) is much slower than for the other two. Therefore, it was concluded that summation inside the weight doesn't lead to any problems.

The CTM A built using weight (2.22) is not enough to compute the free energy or any other observable values. To reconstruct the whole lattice as a power of some CTM, an additional CTM is needed. It could be constructed from A using an auxiliary HRTM G, which in turn is made up from weights

$$w'(a,b,c,d) = \exp\left[\frac{\beta}{2}(ac+bd) + \frac{H}{4}(a+b+c+d)\right]. \quad (2.25)$$

A very simple second auxiliary weight is also required

$$w''(a,b,c,d) = \exp\left[\frac{\beta}{2}(ab+bd) + \frac{H}{4}(a+b+c+d)\right]. \quad (2.26)$$

The algorithm steps are the same as for the square lattice, but an auxiliary matrix G is built from the weight (2.25) using the same projection and diagonalisation matrices U, so the bases of A, F and G are kept consistent.

After iteration one constructs the "whole-lattice" picture in the following manner: Using formula (2.2) the extended CTM is built from A, F and w. With the same formula, an auxiliary CTM is build from A, F and w'', as shown in Figure 2.6. The product of these CTMs is a new non-symmetrical CTM, which repre-

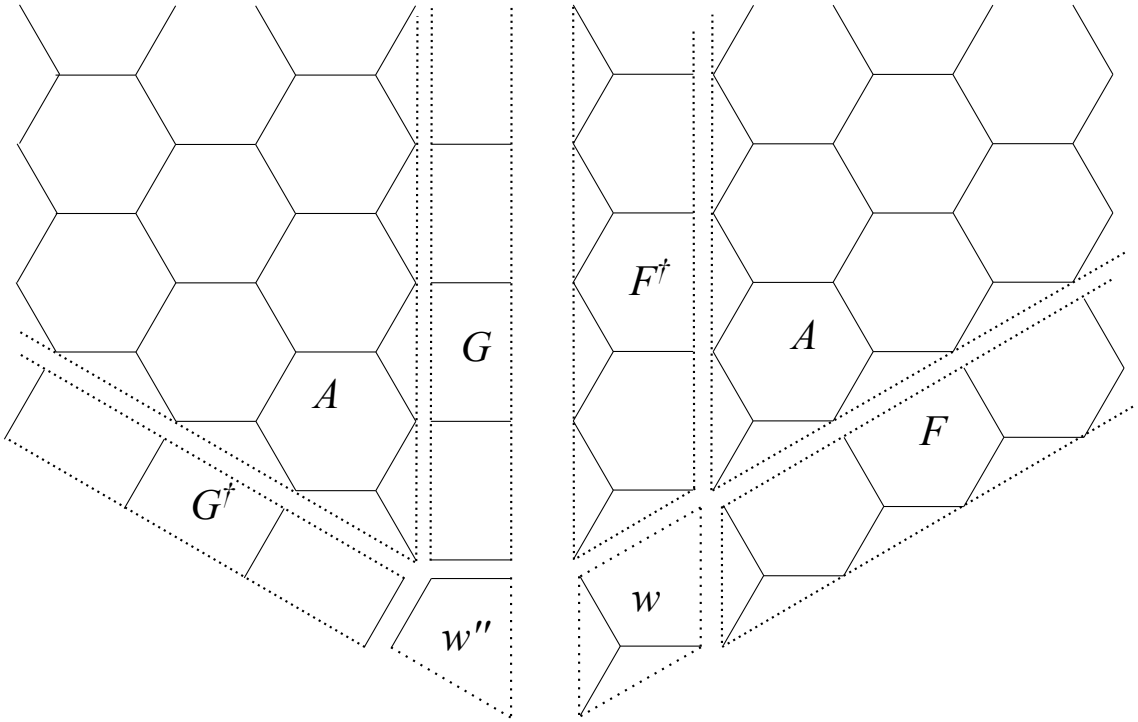

Figure 2.6: The "auxiliary matrix" algorithm for the honeycomb lattice.

sents one third of the lattice and could be directly used in the formula (1.48). Formulas for the free and internal energy are far more complicated than for the square lattice and not worth writing down here.

Problems with the algorithm. While having the least computational overhead in comparison to usual CTMRG, this algorithm failed to produce correct results for $H > 0$. Also the error for the $H = 0$ magnetisation was much larger than for other algorithms and was strongly "levelled" against N, i.e., the error decreased only when a new whole level was added to the truncated $A(a)$, and not smoothly as in any other algorithms used.

The mentioned correspondence between the CTM and the density matrix was attributed to the error. While in any other case the CTM spectrum does represent the density matrix of some associated quantum system, in the case of this algorithm the correspondence is spoiled by the presence of the auxiliary matrices G, and similarly the CTM B does not have a corresponding structure in quantum mechanics. From the "lattice" point of view there are no flaws in the algorithm.

"Big-block" algorithm. This is the most straightforward generalisation of the square-lattice algorithm. The following Boltzmann weight is used

$$w_{abcd}^{efgh} = \sum_{ijk} \exp\left[\frac{\beta}{2}(af + be + bg + hd + 2fi + 2ei + 2ij + 2jk + 2cj + 2kg + 2kh) + \frac{H}{6}(a + 3e + 2b + 3f + 6i + 6j + 6k + 3g + 2c + 3h + d)\right], \quad (2.27)$$

see Figure 2.7.

Unlike the previous algorithm, a single Boltzmann weight can reconstruct the whole lattice, so no additional tricks are needed. As usual, bonds which are shared between two weights are split evenly among those weights. The "field" contribution of a spin in the weight is proportional to the "angle" of the spin in the weight.

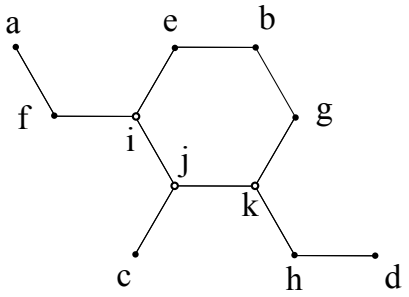

Figure 2.7: The Boltzmann weight in the "big-block" algorithm.

All iteration details are pretty much the same as for the square lattice, the only major difference being that many matrices are now larger than in the usual algorithm, because before diagonalisation 2 spins (instead of 1) are explicitly enumerated. The spin which is close to the corner of the CTM is naturally chosen to have higher weight, i.e., while enumerating all spins it takes one value in the upper half of the extended CTM, and another value in the lower part of the extended CTM, and similarly for the left and right halves for the spin on the other side of the CTM.

As a result of enlargement, diagonalisation of the extended CTM $H(a)$ takes approximately 8 times longer than the usual algorithm, since the matrix diagonalisation algorithm scales as $O(N^3)$ in general. Calculation of observable values is largely the same as for the usual square-weight algorithm, only the sizes of the matrices are different.

"Asymmetric CTM" algorithm. This algorithm uses a non-symmetric Corner Transfer Matrix and diagonalises it using Singular Value Decomposition (SVD). Since it has subtle differences in comparison to the other variants, and it was used for most of the actual calculations, we present it in detailed form.

Indices denoted by capital letters ($I = \{i_1, i_2, \ldots\}$, $J = \{j_1, j_2, \ldots\}$ etc.) are multi-indices and enumerate states of the spins in the basis obtained at the previous iteration steps. These are an "implicit" basis, as opposed to the "explicit" basis where each spin corresponds to its own index. In the Figures, the implicit-basis spins are drawn as open circles, and the explicit-basis spins as filled circles. The total dimension of the implicit-basis space is N.

The steps of the algorithm are:

1. The extended CTM is constructed using the (diagonal) CTM and two identical HRTMs. Note that in this algorithm both HRTMs enter the CTM non-transposed, as

$$H_{iI}^{jJ}(\alpha) = \sum_{\beta I' J'} w_{\alpha i}^{j\beta} \cdot F_{\beta J'}^{jJ} A_{I'}^{J'}(\beta) F_{iI}^{\beta I'}, \tag{2.28}$$

   or in matrix form as

$$H_{ji}(\alpha) = \sum_{\beta} w_{\alpha i}^{j\beta} \cdot F j_{\beta} A(\beta) F_{\beta i}, \tag{2.29}$$

   see Figure 2.8. Here the Boltzmann weight w is

$$w_{cd}^{ab} = \exp(K/2(ab{+}bd{+}cd){+}H/6\,(a{+}2b{+}c{+}2d)) = \left(\begin{array}{c} \text{a} \\ \text{H/6} \quad \text{K/2} \\ \text{b} \\ \text{H/3} \\ \text{K/2} \\ \text{H/3} \\ \text{d} \\ \text{H/6} \quad \text{K/2} \\ \text{c} \end{array} \right). \tag{2.30}$$

2. SVD is performed for $H(\alpha)$ matrix as

$$H(\alpha) = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix} \cdot \begin{pmatrix} D_{11} & \\ & D_{22} \end{pmatrix} \cdot \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix}^{\dagger}. \tag{2.31}$$

   Note that every element on the right side depends on α. The singular values in D are sorted in descending order. The new CTM is obtained from the

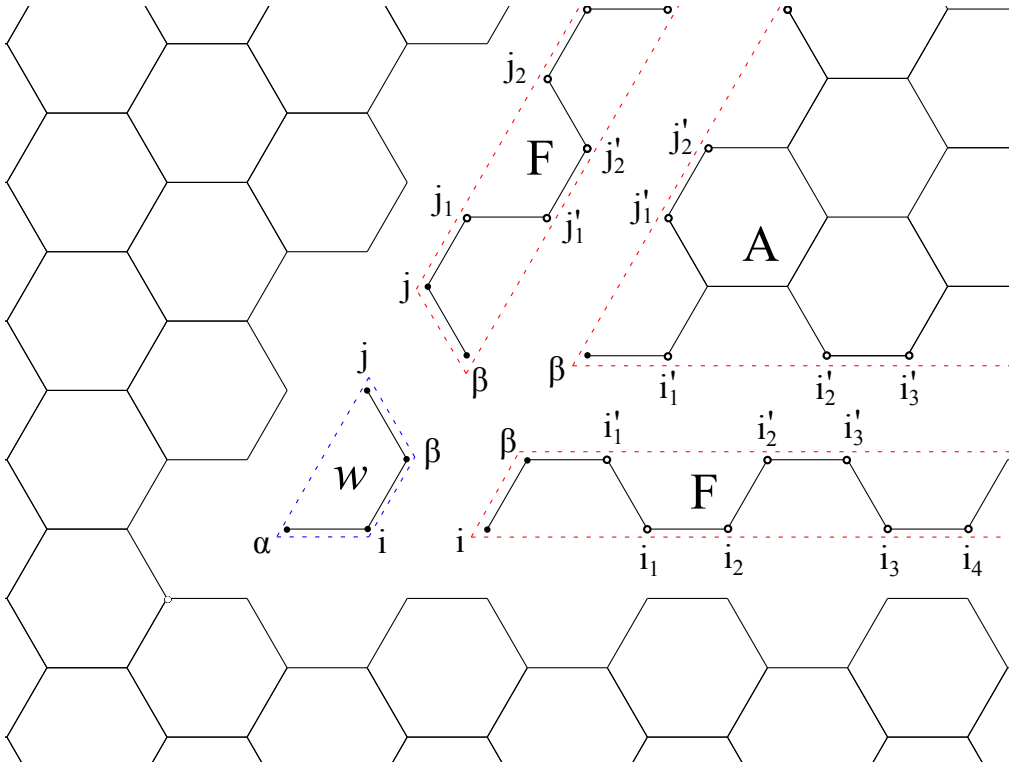

Figure 2.8: Construction of extended CTM, equation (2.28) or (2.29).

largest singular values, i.e. the first half of D:

$$\tilde{A}(\alpha) = D_{11}(\alpha). \tag{2.32}$$

Matrices V and W are truncated consistently with D when they are used to convert the HRTM into the new basis, and both V and W have the same number of columns as the size of D.

For the SVD the "divide-and-conquer" algorithm [44] was used through the `DGESDD` function of LAPACK [51].

3. The extended HRTM is built by attaching an extra Boltzmann weight to the transposed HRTM from the previous iteration:

$$F^{ia}_{jb} = w^{ji}_{ba} F^{\dagger}_{ba}. \tag{2.33}$$

4. The HRTM is then truncated and converted to the new basis:

$$\tilde{F}_{ij} = \begin{pmatrix} W_{11}^{\dagger}(i) & W_{21}^{\dagger}(i) \end{pmatrix} \cdot \begin{pmatrix} F_{j1}^{i1} & F_{j2}^{i1} \\ F_{j1}^{i2} & F_{j2}^{i2} \end{pmatrix} \cdot \begin{pmatrix} V_{11}(j) \\ V_{21}(j) \end{pmatrix}, \tag{2.34}$$

see Figure 4.

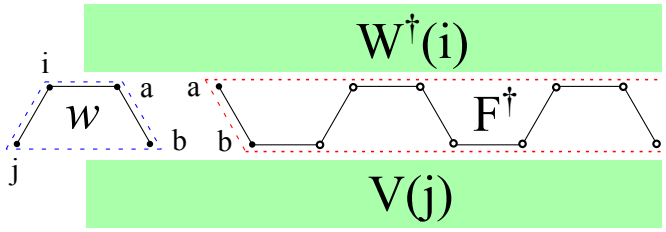

Figure 2.9: Extending HRTM and converting it to a new basis, equations (2.33) and (2.34).

5. Back to the step 1.

It is easy to see that the new bases for A and F are consistent. Consider H and write it with D and $\tilde{F}$ (the double indices and central spin are omitted for clarity)

$$FAF \to (W\tilde{F}V^{\dagger})(VDW^{\dagger})(W\tilde{F}V^{\dagger}) = W(\tilde{F}D\tilde{F})V^{\dagger}. \tag{2.35}$$

The partition function Z is expressed as

$$Z \sim \sum_{\alpha} \text{ tr } \left[A(\alpha)A^{\dagger}(\alpha)\right]^3, \tag{2.36}$$

which is obviously invariant under the basis transformation (2.35).

Formulas for observable values The formula for the magnetisation

$$M = \langle \alpha \rangle = \frac{r_+ - r_-}{r_+ + r_-} \tag{2.37}$$

is trivial, where

$$r_\alpha = \text{ tr } \left[A^{\dagger}(\alpha)A(\alpha)\right]^3 = \text{ tr } A(\alpha)^6. \tag{2.38}$$

The partition function per site is

$$\kappa = \left[\frac{r_1 r_3}{r_2}\right]^{\frac{1}{6}} \tag{2.39}$$

where

$$r_1 = \sum_a \text{ tr } \left[A(a)^\dagger A(a)\right], \tag{2.40a}$$

$$r_2 = \sum_a \text{ tr } \left[\sum_b F(b,a)^\dagger A(b)^\dagger A(b) F(b,a)\right]^3, \tag{2.40b}$$

$$r_3 = \sum_a \text{ tr } \left[H_{iI}^{jJ}(a)\right]^3, \tag{2.40c}$$

and where H_{iI}^{jJ} is from (2.29).

The internal energy per bond is

$$U = \frac{2}{3}\langle -\log w_b \rangle = \frac{r_3'}{r_3} \tag{2.41}$$

where the modified weight

$$w_b = \exp \frac{\beta}{2}(ab + bd + dc) \tag{2.42}$$

only includes bond energies but not the "magnetic" energy, and r_3' is as in (2.40c) with w replaced by w_b. The $\frac{2}{3}$ factor directly follows from Table 1.1. Computing the internal energy with the modified weight w_b is a simple alternative to using the relations between various normalisations of the internal energy, such as U and U' from (1.49) and (1.50).

As usual, one could estimate convergence of the algorithm by computing a simple observable value (usually the magnetisation) and using it as a convergence parameter; i.e., comparing it with itself at previous steps or (for values of parameters with known magnetisation) with the exact value. In the latter case the estimate can be used for parameter values which are further away from the critical point, but don't have the exact solution, e.g., for the Ising model we can use the estimated number of steps obtained while $H = 0$ for the $H \neq 0$ case with all other parameters held the same. Additional tests for the honeycomb lattice consist of comparison of the susceptibility $\chi = \frac{\partial M}{\partial H}$ with recent results of Chan et al. [33].

2.3 Other Related Algorithms

The most obvious competitor to the CTMRG algorithm is the usual Transfer Matrix (TM) algorithm. It was used for many models, and many of the exact results used in this thesis were obtained using the TM, and in fact it is one of the most direct methods of figuring out properties of lattice models. But it obviously has many deficiencies. We name only some of them. The size of the TM depends exponentially on the size of the system, so modelling of systems larger than a dozen sites or so is not possible. Also the spectrum of the TM is much less manageable than that of the CTM, i.e., for zero field the next to ground state level is highly degenerate with a degree of degeneracy depending on the number of sites. Whereas for the CTM, the degeneracy, if any, depends only on the number of level, and not on the number of sites in the system.

Another branch of related algorithms relates to developments of White's algorithm for the needs of quantum mechanics. Some of the algorithms could also be adapted to statistical mechanics models, but most of them are specially designed to work with a quantum-mechanical ψ-function. Some of the most discussed variations of the DMRG are MPS and MERA methods [43, 52, 53, 54].

MPS means "Matrix Product State", and it is very similar to the Ansatz (1.27), but represents not half of the lattice, but a ψ-function of a quantum system. A modification of White's algorithm is used to update the state. Before trying CMTRG we evaluated this algorithm for the Heisenberg chain, although no full-scale tests were done.

A more advanced algorithm is MERA. MERA means "Multiscale Entanglement Renormalisation Ansatz" and it is a development of MPS. Instead of using a "linear" Ansatz, the state is organised as a hierarchical product of matrices. Namely, the state is a fully balanced binary tree. Each level is represented by the same matrix, so the number of matrices required to represent the state is $\log_2 N$, where N is the number of spins. There is evidence that such a representation is very robust in some cases where entanglement is very strong.

One of the generalisations of that method to two-dimensional quantum systems is called PEPS ("Projected Entangled Pair States") [55] and it is of particular interest in statistical mechanics. It seems that under some conditions PEPS could be converted to an algorithm for 3D classical statistical mechanics systems. Since there is no universal generalisation of CTMRG to three dimensions yet, PEPS could help to derive one.

The main problem with these algorithms is that while they might be better

than the usual Hamiltonian diagonalisation near the critical point, the error introduced by the iteration procedure is allegedly rather big, so the methods are not precise enough to obtain the scaling function. Also the derivation of MERA and PEPS don't produce rigorous equations that iterative methods could actually solve, so tracking and explaining the error introduced by the Ansatz could be very difficult. The methods also have some problems with the representation of infinite systems. Nonetheless, they can represent very large systems, much larger than usual Transfer Matrix (or rather, Hamiltonian) methods allow.

Chapter 3

Scaling of the Ising Model

In this chapter we mostly follow results of Fonseca and Zamolodchikov [17], whose work is a very comprehensive study of the analytic structure of the IFT scaling function. Since their work concerns only field theory results, a "bridge" between field theory and lattice models is needed. We describe known theoretical facts required for a basic understanding of the procedure required to extract a lattice-independent scaling function from the lattice calculations.

3.1 Scaling and Universality Hypotheses

As mentioned in the Introduction, scaling in statistical mechanics is one of the very simple, but fundamental concepts. Scaling is manifest in many physical systems, even in very simple mechanical ones. For example, if we have a conservative mechanical system, then we can change the scales of all lengths, masses, forces etc, and obtain a scaled system, the system where certain ratios of the physical quantities will be the same. Each value must be scaled according to its dimension, and unchanged quantities are dimensionless quantities.

In the infinite systems an additional property could emerge. The system with changed parameters could be in principle totally identical to the original system, i.e. the system could be invariant under the scaling transformation. This is called scale invariance. In lattice statistical mechanics that situation could be realised when the natural scale of the system – its correlation length – becomes infinite. This happens at the critical point.

Mathematically, all this means that the free energy is a homogeneous function

of its arguments:

$$F(a, b, c, \ldots) = \mu^{y_0} F(a\mu^{y_a}, b\mu^{y_b}, c\mu^{y_c}, \ldots). \tag{3.1}$$

Here $\mu \sim 1$ is a scale change, y_a, y_b, y_c are scaling dimension and y_0 depends on the dimension of the system. We can now reduce the number of required parameters by using dimensionless combinations p_1, q_1, ... of the parameters a, b, c ... as arguments to the homogeneous function

$$F(a, b, c, \ldots) \sim \tilde{F}_1(p_1, q_1, \ldots) \tag{3.2}$$

For the Ising model, where the parameters are H and T, the reduced function will depend only on one dimensionless parameter. The corresponding function $\tilde{F}$ is called the scaling function.

For the Ising model another possibility is realised. The free energy contains logarithmic singularity and F has the form

$$F(a, b, c, \ldots) = F_1 \log F_2 + F_3 \tag{3.3}$$

where F_1 is a regular function. If we rescale F_2 then some regular term adds to the F_3 term, but the regular part of F_3 becomes negligible in comparison to the singular part of F_3, if we come close enough to the critical point.

In statistical mechanics, one usually considers such parameters where $y_i > 0$, i.e., values that become more significant if we scale the system up. Parameters with $y_i < 0$ are called irrelevant fields in the QFT approach. They will be mentioned later in this chapter, but omitted almost everywhere else.

Since the various physical quantities could be obtained as derivatives of F with respect to the parameters $a, b, c \ldots$, these physical quantities will also have some scaling properties, which actually were originally studied. Asymptotic behaviours of these quantities in the vicinity of the critical point follow a power law, and the various powers are called critical indices.

3.1.1 Critical Indices

One of the most important properties of the statistical models with phase transitions are the critical indices of these transitions. They show the degree of the leading terms near the critical singularity. For lattice models similar to the Ising Model the following critical indices are usually considered:

- α – this index describes the asymptotic behaviour of the specific heat $c = \frac{\partial U}{\partial T}$ in the vicinity of the critical point when $H = 0$:

$$c(T, 0) \sim |T - T_c|^{-\alpha}. \tag{3.4}$$

  If the asymptotics of the specific heat differ for $T > T_c$ and $T < T_c$, then different exponents α and α' are considered, but this is not the case for the Ising model. However, the Ising model represents another "special" case, since it has a logarithmic singularity in the specific heat, so $\alpha = 0$:

$$c(T, 0) \sim \log |T - T_c|. \tag{3.5}$$

  The presence of a logarithmic singularity requires special treatment, because it is usually assumed that the scaling function can be expressed in terms of a series, and a logarithmic singularity cannot be represented in that way.

- β describes the asymptotics of the spontaneous magnetisation when $T \to T_c^-$:

$$M(T, 0) \sim (T_c - T)^{\beta}, \quad T < T_c,\ H = 0 \tag{3.6}$$

  For the Ising model, $\beta = 1/8$.

- γ describes the asymptotics of the susceptibility, again for $H = 0$:

$$\chi \sim |T - T_c|^{-\gamma}. \tag{3.7}$$

  As for α, there could be different values of this exponent in different phases, but for the Ising model both the magnetic and disordered phases have $\gamma = 7/4$.

- δ describes the asymptotics of the magnetisation. For non-zero field and on the critical isotherm $T = T_c$:

$$M(T_c, H) \sim |H|^{1/\delta}. \tag{3.8}$$

  For the Ising model $\delta = 15$.

There are several other exponents, some of which are applicable to the Ising model, but they are not of interest here.

These indices are not independent. Some of the simplest quantitative results that one could obtain from the scaling hypothesis are relations between some of

these indices [10, 37, 20]:

$$\gamma = \beta(\delta - 1), \tag{3.9}$$

$$\alpha + 2\beta + \gamma = 2. \tag{3.10}$$

3.1.2 Ising Model Scaling Function

For the Ising model the dimensions of the relevant fields are

$$y_\tau = \frac{1}{2}, \tag{3.11}$$

$$y_H = \frac{15}{16}. \tag{3.12}$$

These numbers directly follow from the values of the critical indices β and δ (3.6), (3.8).

One of the most striking practical manifestations of scaling is the dependency of $F_{\text{sing.}}$ on a dimensionless combination of these parameters, with the number of combinations one less than the number of parameters. That is, for the Ising case, near the critical point the singular part of the free energy is expressed as a function of only one variable. The variable is non-unique, but there are two natural choices:

$$\eta = \frac{\tau}{|H|^{8/15}} \quad \text{for } H \neq 0, \tag{3.13}$$

$$\xi = \frac{H}{|\tau|^{15/8}} \quad \text{for } \tau \neq 0, \tag{3.14}$$

where the factors will be discussed later. These variables are called scaling variables, and a part of the partition function that depends on one of them is a "scaling function", the main topic of this work.

In (3.13) and (3.14) we have simply $\tau \sim T - T_c$ without any additional connotations, but since for $H = 0$ we have $T > T_c$ vs. $T < T_c$ duality expressed via τ from (1.13a), (1.13b) and (1.13c) (which are, obviously, $\tau \sim T - T_c$), it is natural to relate τ from Chapter 1 to τ from formulas above. Later in this chapter that correspondence is described more thoroughly.

For the Ising case we have already mentioned an additional difficulty, which is not taken into account by (3.1). The scaling law can be extended by adding terms with logarithms to the power series expansion of the scaling function as in (3.2). This is exactly the situation for the Ising model, since it has critical index

$\alpha = 0$, which means a logarithmic singularity for the specific heat. Very roughly, the total singular part of the free energy could be written as:

$$F_{\text{sing.}}(\tau, H) \sim \tau^2 \log \tau + F_{\text{ser.}}(\tau, H) \tag{3.15}$$

where $F_{\text{ser.}}$ can be expanded as a power series.

Note that there could be more terms with logarithms, but (3.15) is chosen solely due to the corresponding Onsager's solution.

3.1.3 Universality Hypothesis

Another hypothesis related to the scaling function of the various statistical mechanics models is the Universality Hypothesis [37, 39]. It is somewhat more difficult to understand than the scaling hypothesis and probably harder to show its validity.

The basic statement of the hypothesis is that all statistical models with scaling behaviours can be divided into universality classes which share not just critical indices, but the whole scaling function. Therefore all members of each class share the same properties near the critical points.

Consider an abstract Hamiltonian consisting of the main part H_0 with an addition H_1:

$$H = H_0 + \epsilon H_1 \tag{3.16}$$

If we have parameter ϵ varying, then the universality hypothesis states that the critical behaviour of the system changes only when the symmetry of the Hamiltonian changes, which can happen when parameter ϵ has some special values.

While the scaling hypothesis allows us to predict critical properties of the system from the scaling function only for a single model, the main point of universality is that knowing the scaling function of one model in a universality class, we can predict the critical behaviour of some other, possibly much more complicated models belonging to the same universality class.

The universality hypothesis gained an additional meaning with the development of Conformal Field Theory. In the original paper describing CFT authors pointed out that properties of the simplest variants of the CFT (namely CFTs with rational central charge) correspond to the critical properties of various 2D lattice statistical mechanics models. The case $c = 1/2$ is the basis for the Ising Field Theory, which corresponds to the vicinity of the critical point of the lattice

model.

While assumptions used to justify the universality hypothesis are rather general, there are cases when the universality hypothesis seems to fail. Baxter [20] highlights that the eight-vertex model is one of the most well known examples of the failure of the universality hypothesis.

3.2 Ising Field Theory

Two-dimensional Conformal Field theory is one of the fundamental models in mathematical physics. It was introduced by Belavin, Polyakov and Zamolodchikov in [11], after more general ideas of conformal invariance in arbitrary dimensions [56], and ultimately from the earlier general scaling concept.

CFT is a two-dimensional quantum field theory of massless interacting particles. There are an infinite number of commuting integrals of motion in the theory. The theory is integrable, so that in principle it is possible to compute exactly any correlation function. Moreover, the procedure for computing the correlation functions is constructive and well defined.

There is a single free parameter c called the central charge that defines the whole spectrum. Among many other 2D systems CFTs with different rational c describe the critical limit of various two-dimensional statistical mechanics models. In [11] it is shown that $c = 1/2$ corresponds to the critical 2D Ising model.

The operator content of the theory consists of operators with conformal weights $1/16$ and $1/2$ that are directly related to magnetic and thermal perturbations. It can be shown that the ground-state energy density F corresponds to the free energy per site of the lattice model. Away from the critical point the model's action is

$$\mathcal{A}_{\mathrm{IFT}} = \mathcal{A}_{1/2} + h \int dV\, \phi_1 + m \int dV\, \phi_2 \tag{3.17}$$

where $\mathcal{A}_{1/2}$ is an action of the usual CFT with central charge $c = 1/2$, ϕ_1 is a primary field with conformal dimension $\Delta = 1/16$ and ϕ_2 is a primary field with $\Delta = 1/2$. This theory is not integrable if both parameters h and m are non-zero. However, if only of one these operators is present, the theory is integrable.

If $h = 0$, then there is a solution corresponding to Onsager's solution, but in the framework of the field theory model. The model consists of free (non-interacting) fermions with a non-zero mass that depends on the value of m. When $m \to 0$ the mass of the fermions becomes zero and the correlation length becomes infinite.

E_8 theory. Another exact solution of (3.17) is possible when $m = 0$. In this case the CFT is perturbed by the external magnetic field h only, corresponding to the critical isotherm $T = T_c$ of the lattice models. It was obtained by Zamolodchikov [18] (see also [57]). The spectrum of the model has several particles, with effective masses corresponding to the root lengths of the exceptional Lie algebra E_8. Solution of the model permits exact calculation of the first two expansion coefficients of the scaling function.

It is worth noting that this solution was recently checked experimentally [58], at least to some extent. In the experimental setup the one-dimensional quantum Ising chain was constructed from cobalt atoms; certain chain properties directly correspond to the properties of the 2D classical statistical mechanics Ising model. The spectrum of the chain excitations was measured by neutron scattering. The ratios of the two lowest excitations (which equals the golden ratio $(\sqrt{5}+1)/2$) was compared with the theoretical prediction and the error was relatively small for a rather direct experiment. In [58] the group is described as "Remarkably, the simplest of the systems, the Ising chain, promises very complex symmetry described mathematically by the E_8 Lie group. [...], the highest order-symmetry group discovered in mathematics, which has not yet been experimentally realized in physics", and the results of the experiment are "This is indeed consistent with this highly non-trivial prediction of two prominent peaks at low energies, which we identify with the first two particles m_1 and m_2 of the off-critical Ising model. [...] the golden ratio m_2/m_1 [...] predicted for the E_8 masses".

3.2.1 Scaling Function of the IFT

We describe the scaling function of the IFT as reported in the fundamental paper [17]. There the scaling function was studied analytically and essentially computed numerically using IFT-only methods. We have used these results as a basis for our fitting procedure, after taking into account the scaling corrections in Section 3.3. The resulting scaling function is used as a trial function for the lattice scaling. In principle it is possible to derive the expansions without any reference to IFT, i.e. purely from the lattice observations, but as with any other artificial limitation it would spoil the precision of the results.

In the IFT the dimensionless scaling variables (analogues of the dimensionless variables introduced in Section 3.1) depend on the parameters m and h and have the form

$$\xi = \frac{h}{|m|^{15/8}}, \tag{3.18}$$

when $m \neq 0$, and

$$\eta = \frac{m}{|h|^{8/15}} \tag{3.19}$$

when $h \neq 0$. When both h and m are not zero (and real) then η and ξ are related simply by

$$\eta = \pm \frac{1}{\xi^{8/15}}, \tag{3.20}$$

with the sign depending on whether η has positive or negative value.

We could express the energy density of the vacuum state of (3.17) via both η and ξ. For areas in (m, h) space where $m \neq 0$ the energy takes the form

$$F(m, h) = \frac{m^2}{8\pi} \log m^2 + m^2 G(\xi). \tag{3.21}$$

When $h = 0$ the expression reduces to the "pure" Onsager singularity

$$F(m, 0) = \frac{m^2}{8\pi} \log m^2. \tag{3.22}$$

For areas where $h \neq 0$ we have

$$F(m, h) = \frac{m^2}{8\pi} \log m^2 + |h|^{16/15} \Phi(\eta), \tag{3.23}$$

which reduces to Zamolodchikov's solution

$$F(0, h) = \Phi_0 |h|^{16/15} \tag{3.24}$$

when $m = 0$.

Obviously, the functions $G(\xi)$ and $\Phi(\eta)$ are not independent and in fact are the single scaling function that is just expressed via different variables. We study $G(\xi)$ and $\Phi(\eta)$ mostly by using their series expansions. The complete analytical structure of the scaling function is rather complicated (and not known in full), and only when taken together the convergence zones of the series corresponding to $G(\xi)$ and $\Phi(\eta)$ cover the whole real axis.

Since the functions $G(\xi)$ and $\Phi(\eta)$ are not independent, their expansion coefficients can be expressed in terms of each other. This procedure is widely used in [17] using dispersion relation formulas. In this work all coefficients are treated as totally independent and are obtained from the different zones in the lattice parameter space (H, T), i.e., from different numerical data.

We now describe the structure of the scaling function representations and

their series expansions. The function $\Phi(\eta)$ is singular at $\eta = 0$, but its singularity must compensate the Onsager singularity, so it has the form

$$\Phi(\eta) = -\frac{\eta^2}{8\pi} \log \eta^2 + \eta^{16/15} \tilde{\Phi}(\eta), \tag{3.25}$$

where the auxiliary function $\tilde{\Phi}(\eta)$ is regular in some domain around $\eta = 0$ and can be represented as a power series:

$$\tilde{\Phi}(\eta) = \Phi_0 + \Phi_1 \eta + \Phi_2 \eta^2 + \Phi_3 \eta^3 + \cdots . \tag{3.26}$$

The radius of convergence of the $\tilde{\Phi}(\eta)$ series is about 2.42. The singularity (the branch cut) is not located on the real axis, so it is not possible to see it directly from our data.

For $G(\xi)$ we actually need to introduce two functions, for $\eta > 0$ and $\eta < 0$, (both for $\xi > 0$) to correctly represent the strikingly different behaviour of the Ising model below T_c and above T_c.

For $T > T_c$ we have a convergent series:

$$G_{\text{high}}(\xi) = G_2 \xi^2 + G_4 \xi^4 + G_6 \xi^6 + \cdots = \sum_{k=1}^{\infty} G_{2k} |\eta|^{-30k/8}, \quad \eta \to -\infty. \tag{3.27}$$

The series is absolutely convergent around $\xi = 0$. The domain of convergence is limited by the Yang-Lee singularities that are located on the imaginary axis and limit the radius of convergence to

$$\xi_0 = 0.18930(5). \tag{3.28}$$

Since it is rather small in comparison to 1, the coefficient G_i can grow rather quickly with i and this is actually the case.

The series expansion for $T < T_c$ ($\eta > 0$) is not absolutely convergent for any $\xi > 0$ because of the cut on the real axis that starts at the origin and goes to $-\infty$. The cut can be seen as a manifestation of the broken symmetry and spontaneous magnetisation. The asymptotic expansion contains all natural powers of ξ:

$$G_{\text{low}}(\xi) = \tilde{G}_1 \xi + \tilde{G}_2 \xi^2 + \ldots = \sum_{k=1}^{\infty} \tilde{G}_k |\eta|^{-15k/8}, \quad \eta \to +\infty. \tag{3.29}$$

Although not an absolutely convergent series, its actual convergence is not very

bad and can be used for the scaling function approximation for most of the $\eta > 2.42$ domain where the $\tilde{\Phi}(\eta)$ expansion doesn't converge.

In [17] some other representations of the scaling function are derived and used, but since we didn't use them in any way in our work, we don't list them here.

Known expansion coefficients. Some of the coefficients in (3.26), (3.27) and (3.29) are known exactly.

The coefficient $\tilde{G}_1$ has been known for a long time [59] and in fact was computed from the lattice model,

$$\tilde{G}_1 = -\frac{2^{1/12}\mathcal{A}^{3/2}}{e^{1/8}} = -1.3578383417065954956\ldots. \tag{3.30}$$

Here $\mathcal{A} = 1.282427\ldots$ is the Glaisher constant.

The coefficients G_2 and $\tilde{G}_2$ are not known in closed form, but can be computed up to arbitrarily high precision using the solutions of the Painlevé III equation. They were formally computed in [31], and in much more manageable form in [32], in the study of the susceptibility of the lattice Ising model. The values of the coefficients are:

$$G_2 = -1.845228078232838354322445111140727\ldots, \tag{3.31}$$

$$\tilde{G}_2 = -0.048953289720274595831520781527701\ldots. \tag{3.32}$$

The number of given digits is already far beyond the precision we could numerically achieve.

For the $\tilde{\Phi}(\eta)$ series two coefficients are known exactly. Φ_0 is obtained directly from Zamolodchikov's solution and its value is

$$\Phi_0 = -(2\pi)^{\frac{1}{15}} \frac{\gamma(\frac{1}{3})\gamma(\frac{1}{5})\gamma(\frac{7}{15})}{\left[\gamma(\frac{1}{4})\gamma^2(\frac{3}{16})\right]^{\frac{8}{15}}} = -1.19773338379799\ldots, \tag{3.33}$$

where

$$\gamma(x) = \frac{\Gamma(x)}{\Gamma(1-x)}. \tag{3.34}$$

Φ_1 was computed as a quadrature in [60] using an extension to Zamolodchikov's solution. In our work [61] we were able to calculate the quadrature in

closed form using the same auxiliary function $\gamma(x)$:

$$\Phi_1 = -\frac{32\ \ 2^{\frac{3}{4}}}{225(2\pi)^{\frac{7}{15}}} \frac{\gamma(\frac{1}{3})\gamma(\frac{1}{8})}{\left[\gamma(\frac{1}{4})\gamma^2(\frac{3}{16})\right]^{\frac{19}{15}}} \prod_{k=3}^{7} \gamma\left(\frac{k}{15}\right) = -0.3188101248906\ldots. \quad (3.35)$$

All other coefficients are known only approximately.

IFT numerical solution. Apart from the study of the analytic structure, there is an extensive numerical study of the scaling function presented in [17].

The method used is the Truncated Free-Fermion Space Approach, which was designed specifically for the IFT. The method is based on the diagonalisation of the Hamiltonian form of (3.17) in the basis of multi-particle states of free fermions. The leading expansion coefficients Φ_i are then extracted from the ground state eigenvalue, while other coefficients are obtained via the dispersion relation with some assumptions about the analytic structure of $F(m, h)$.

There are several other studies of the scaling function expansions which are however not as complete as [17]. The coefficient G_4 was calculated in [62] with high accuracy using a form-factor expansion of the four-point correlation function. In another paper by the same group, an "equation of state" of the near-critical Ising model was devised, which yielded values of G_i [63].

3.3 Lattice Corrections to Scaling

The scaling hypothesis holds exactly at the critical point where the correlation length is infinite. To actually "experience" scaling predictions we need to vary parameters and move away from the critical point. Away from criticality the correlation length is finite and thus the properties of the lattice come into play.

There are two notable contributions of the lattice effects to scaling. One is to the regular part of the free energy, the other is to the lattice scaling fields. Both contributions grow steadily when we move away from the critical point. To calculate the scaling function we must use all our knowledge of them and keep the balance between numerical precision and partially unknown lattice contributions. To have a better chance to fit singular terms away from the critical point we need to have more precise calculations (i.e., more digits in our numbers), which is computationally expensive if we go beyond QUAD precision. Hence it is crucial to find a "balance" between two factors that make it difficult to "extract" the scaling function: bad precision of calculations very close to the critical point due

to large correlation length, and contribution from the lattice terms (the regular part and the lattice fields) further away from the critical point.

There is also one "smaller" correction, which is much less well known and understood. There are irrelevant fields in the equation (3.1). In most cases they are not considered, since their contribution don't matter in the ultimate statistical mechanics problems, i.e., their influence on the infinite systems is negligible. We don't try to reveal the presence of the irrelevant IFT operators. The only exception is a fit of the square lattice susceptibility $\chi(\tau)$, where it is possible to get some information about the irrelevant operators and confirm their CFT weight.

3.3.1 Regular Part of the Free Energy

The first lattice contribution is fairly well understood. In many lattice models the free energy splits into parts with different analytical properties at the critical point. For the Ising model we have

$$F_{\text{total}}(H,T) = F_{\text{reg.}}(H,T) + F_{\text{sing.}}(H,T) + F_{\text{sub.}}(H,T), \quad H \to 0,\, T \to T_c. \tag{3.36}$$

Here $F_{\text{sing.}}$ is the part that contains the leading singularity associated with the scaling law discussed in Section 3.1. Only this part is present in the IFT, as mentioned in Section 3.2.

However, on the lattice there is a contribution to the total free energy which is a regular function even at the critical point. It doesn't follow any scaling laws. Since it is regular, its derivatives around the critical point are much smaller than the unbounded derivatives of the singular part, except for the first few derivatives of $F_{\text{sing.}}$ which are bounded. For example, for the Ising model the susceptibility χ and the specific heat c are unbounded, and one could neglect the regular part close enough to the critical point.

But we cannot approach too close to the critical point while maintaining good precision, and we also analyse lower bounded derivatives in Zone 2 (see Section 4.1). So we carefully treat the regular part and try to take its contribution into account.

The third term $F_{\text{sub.}}$ is a singular term, but with singularity of lesser degree than $F_{\text{sing.}}$. It is associated with irrelevant operators. In practise it is usually quite small in value and will be discussed later in Section 3.3.3.

By definition the regular part can be expressed as a convergent series near the

critical point:

$$F_{\text{reg.}}(H,\tau) = A(\tau) + B(\tau)\cdot H^2 + O(H^4), \quad H\to 0,\ \tau\to 0. \tag{3.37}$$

Here we have only included the terms that we track later. Note that $B(\tau)$ from (3.39) and $B_\chi(\tau)$ from (1.21) are different functions and should not be confused. In the later formulas they are referred to mostly as expansion coefficients, so no mix up should occur.

The spontaneous magnetisation of the finite lattice Ising model is zero, and the free energy is even with respect to H. When we take the limit of the infinite lattice, some part of the finite lattice free energy becomes the singular part $F_{\text{sing.}}$ and changes its symmetry, and the non-zero spontaneous magnetisation M appears. Since the regular part stays regular while taking the limit, it is still even with respect to H for the infinite lattice.

We use separate coefficients from the expansion of $A(\tau)$ and $B(\tau)$, so we explicitly write down the series for them,

$$A(\tau) = A_0 + A_1\tau + A_2\tau^2 + A_3\tau^3 + \cdots, \tag{3.38}$$

and

$$B(\tau) = B_0 + B_1\tau + B_2\tau^2 + B_3\tau^3 + \cdots. \tag{3.39}$$

Using Onsager's solution of the Ising model, one can explicitly calculate the $A(\tau)$ part of $F_{\text{reg.}}$. We summarise the first few terms of $A(\tau)$ for all three lattices under consideration:

$$A^{(s)}(\tau) = -\frac{\mathcal{G}}{\pi} - \frac{\log 2}{2} + \frac{1}{2}\tau - \frac{1+5\log 2}{4\pi}\tau^2 - \frac{1}{12}\tau^3 + \frac{5(1+6\log 2)\tau^4}{64\pi} + O(\tau^5), \tag{3.40a}$$

$$A^{(t)}(\tau) = -\frac{5}{2\pi}\mathrm{Cl}_2\left(\frac{\pi}{3}\right) - \frac{1}{4}\log\frac{4}{3} + \frac{1}{3}\tau - \left(\frac{2+3\log 12}{8\pi\sqrt{3}} - \frac{1}{36}\right)\tau^2 - \frac{7}{648}\tau^3 + \left(\frac{4+9\log 12}{288\pi\sqrt{3}} - \frac{1}{324}\right)\tau^4 + O(\tau^5), \tag{3.40b}$$

$$A^{(h)}(\tau) = -\frac{5}{4\pi}\mathrm{Cl}_2\left(\frac{\pi}{3}\right) - \frac{1}{4}\log 12 - \frac{1}{3}\tau - \left(\frac{2+\log 432}{16\pi\sqrt{3}} + \frac{1}{36}\right)\tau^2 - \frac{7}{648}\tau^3 + \left(\frac{1}{324} - \frac{\log 8 - 8\log 12 - 4}{576\pi\sqrt{3}}\right)\tau^4 + O(\tau^5), \tag{3.40c}$$

where $\mathcal{G} = \mathrm{Cl}_2\left(\frac{\pi}{2}\right) = 0.915965594\ldots$ is Catalan constant, $\mathrm{Cl}_2(x)$ is Clausen's

integral

$$\mathrm{Cl}_2(x) = -\int_0^x \log\left[2\sin\frac{t}{2}\right]\,dt, \tag{3.41}$$

with $\mathrm{Cl}_2\left(\frac{\pi}{3}\right) = 1.014941606\ldots$.

As there is no exact solution for $H \neq 0$, no exact expansion of $B(\tau)$ (or any higher terms) is known. A semi-analytic form of $B(\tau)$ could be obtained from the susceptibility results, and that was the case for the square lattice. At least the leading coefficient B_0 could be determined entirely from our data using (4.4). And as it turns out, all other coefficients are not needed in the fitting series that we used.

3.3.2 Aharony-Fisher Non-Linear Lattice Fields

Aharony and Fisher [64, 65] suggested another lattice correction to the basic scaling law. The idea consists of replacing the lattice variables with non-linear scaling fields, which are regular functions of the real lattice parameters near the critical point:

$$F_{\text{sing.}}(H,\tau) = \mathcal{F}(h(\tau,H), m(\tau,H)). \tag{3.42}$$

Here $\mathcal{F}(h,m)$ is the universal scaling function and functions $m(\tau,H)$ and $h(\tau,H)$ admit series expansions:

$$m(\tau,H) = -C_\tau \tau a(\tau) + H^2 b(\tau) + O(H^4), \quad \tau \to 0,\ H \to 0, \tag{3.43}$$

and

$$h(\tau,H) = C_h H[c(\tau) + H^2 d(\tau) + O(H^4)], \quad \tau \to 0,\ H \to 0. \tag{3.44}$$

Note that the different way the factors C_τ and C_h enter the expansions is just a convention without any profound physical meaning.

The main reason for the introduction of the non-linear functions is to extend the small region where scaling theory is applicable. The original scaling theory could be seen as a linearisation of the modified one. Whilst the domain of the theory's applicability is still not rigorously defined, it is clear that it is enlarged by introducing these functions. The functions are apparently non-universal and we need to determine them in order to obtain the universal form of $\mathcal{F}(h,m)$.

We shall call these variables "lattice scaling fields", since they directly correspond to the primary fields m and h of the IFT. The coefficients C_h and C_τ,

which are arbitrary if we consider just scaling of the single model, are uniquely fixed by comparing $\mathcal{F}(h,m)$ from the IFT and the lattice $\mathcal{F}(h,m)$.

We write down expansions of the fields for later use:

$$a(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 + \cdots , \tag{3.45}$$

and

$$c(\tau) = c_0 + c_1\tau + c_2\tau^2 + c_3\tau^3 + \cdots . \tag{3.46}$$

These functions can be calculated exactly, together with the leading coefficients C_τ and C_h.

Standard notation for the coefficients of the H^2 terms are

$$b(\tau) = \mu_h(1 + b_1\tau + O(\tau^2)), \tag{3.47}$$

and

$$d(\tau) = e_h + O(\tau). \tag{3.48}$$

In the two latter series we wrote down only the coefficients that we are working with, and which can be determined solely from our numerical data, except for b_1 which could be obtained as an exact number from the susceptibility data for the square lattice. For now they are treated as unknown constants.

The calculation of $a(\tau)$ and $c(\tau)$ is done by comparison of the leading singularities in the lattice and IFT solution. On the IFT side the basis for the comparison is the scaling function with $h = 0$:

$$\mathcal{F}(0,m) = \frac{m^2}{8\pi}\log m^2, \tag{3.49}$$

while the leading singularity in Onsager's solution is on the lattice side. After investigating the singularity in (1.12a), (1.12b), (1.12c) and considering the coefficient coupled to the $\log\tau$ term, one could find $a(\tau)$. For the square lattice, an expression is known in closed form [66]:

$$a(\tau) = \left[\int_0^1 dx \frac{F(1/2,1/2,1;-x\tau^2)}{(1+x\tau^2)^{1/2}}\right]^{1/2}, \tag{3.50}$$

where $F(a,b,c;z)$ is the Gauss hypergeometric function.

For the triangular and honeycomb lattice only the expansion form is known,

and $a(\tau)$ is the same for both lattices. The expansions of $a(\tau)$ are

$$a^{(s)}(\tau) = 1 - \frac{3}{16}\tau^2 + \frac{137}{1536}\tau^4 + O(\tau^6), \tag{3.51}$$

$$a^{(t)}(\tau) = a^{(h)}(\tau) = 1 - \frac{1}{24}\tau^2 + \frac{47}{10368}\tau^4 + O(\tau^6). \tag{3.52}$$

Using the same calculation, the leading coefficients C_τ are determined as

$$C_\tau^{(s)} = \sqrt{2}, \quad C_\tau^{(t)} = \frac{\sqrt{2}}{\sqrt[4]{3}}, \quad C_\tau^{(h)} = \frac{1}{\sqrt[4]{3}}. \tag{3.53}$$

To calculate $c(\tau)$ and C_h we consider the expression for the spontaneous magnetisation in terms of the τ variable,

$$M(\tau) = (1 - k(\tau)^2)^{1/8}, \quad \tau < 0\,(T < T_c), \tag{3.54}$$

and compare it with the leading term in (3.29). This leads to an expression for $c(\tau)$:

$$C_h\tilde{G}_1 \cdot c(\tau) = \frac{M(\tau)}{(-C_\tau \tau a(\tau))^{1/8}}, \tag{3.55}$$

where C_h is determined for the leading constant coefficient of $c(\tau)$ to be 1.

The linear scaling fields coefficients for the lattices under considerations are

$$C_h^{(s)} = -\frac{2^{3/16}}{\tilde{G}_1}, \quad C_h^{(t)} = -\frac{2^{5/16}}{3^{3/32}\tilde{G}_1}, \quad C_h^{(h)} = -\frac{2^{3/8}}{3^{3/32}\tilde{G}_1}. \tag{3.56}$$

Corresponding expansions of $c(\tau)$ are

$$c^{(s)}(\tau) = 1 + \frac{1}{4}\tau + \frac{15}{128}\tau^2 - \frac{9}{12}\tau^3 - \frac{4333}{98304}\tau^4 + O(\tau^5), \tag{3.57}$$

and

$$c^{(t)}(\tau) = c^{(h)}(\tau) = 1 + \frac{1}{6}\tau + \frac{5}{96}\tau^2 + \frac{1}{576}\tau^3 - \frac{727}{165888}\tau^4 + O(\tau^5). \tag{3.58}$$

Since $k(\tau)$ as a function of τ is the same for the triangular and honeycomb lattice, the $c(\tau)$ are also the same.

3.3.3 Irrelevant Operators

In the previous section some part of the scaling function was designated as $F_{\text{sub.}}$. Roughly speaking, it contains all terms that are not regular, and could not be a part of $F_{\text{reg.}}$, and at the same time don't follow a simple scaling law like $F_{\text{sing.}}$. They also exhibit a smoother singularity than $F_{\text{sing.}}$. As it also turns out, the amplitude of the most significant terms in $F_{\text{sub.}}$ is quite small, so it is rather difficult to show even the existence of the first term, let alone existence of higher degree terms. Nevertheless we were able to do this at least in the case of the square lattice, where additional data from [32] allowed us to subtract more, and also the leading degree $F_{\text{sub.}}$ is lower than for the triangular lattice.

For the Ising model the most noticeable part of $F_{\text{sub.}}$ corresponds to the irrelevant operators, i.e., operators that scale with dimension $y_i < 0$ (see Section 3.1). The initial Aharony-Fisher conjecture concerning the non-linear scaling fields suggested that there are no irrelevant operators. After the operators were found using several independent methods, it was established beyond any doubt that they contribute to the scaling on the lattice, and that they are non-universal. In a series of papers by Vicari et al. [62, 67, 63, 19] a hypothesis is stated that certain IFT operators could contribute to the $F_{\text{sub.}}$ term.

The main statement of the hypothesis is that quasiprimary operators with conformal spin being a multiple of the lattice coordination number "survive" on the lattice and give a contribution to the free energy. Quasiprimality means that operators behave like primary operators, but only under the action of a small conformal group (generated by L_{-1}, L_0, L_1), and not under the whole Virasoro algebra. It is required that the contribution of the operators be translationally invariant.

The full analysis of the possible contribution is outlined and confirmed numerically in [19], with more recent semi-analytical lattice calculations in [33]. Due to the complexity of the actual expressions of quasiprimary operators via Virasoro algebra generators and other complexities, we don't present any analysis in this thesis. Moreover, the basic analysis predicts only the order where contributions are possible and doesn't tell us anything about the amplitude of the contribution. In the susceptibility series for the square lattice it was shown that the most significant contribution occurs at fourth order in (1.20), and for the honeycomb and triangular lattices the contribution is at sixth order.

Chapter 4

Numerical Results

In the previous chapter the theory behind scaling and universality was described to an extent allowing us to extract the unknown scaling function from the observable values like $M(H,T)$ and $U(H,T)$. To actually derive the scaling function $\Phi(\eta)$ extensive lattice calculations of M and U are required, followed by fitting Φ_i (as well as G_i and $\tilde{G}_i$) coefficients and unknown values from the regular part of the free energy and Aharony-Fisher scaling fields. The procedure is largely the same for all three types of lattice considered, the key differences being the availability of more data for the square lattice and the irrelevant operators' contribution.

4.1 Details of the Calculations

Calculation for each particular lattice consists of several thousand points with CTM size equal to 128. For the largest CTM size, calculations with QUAD precision are performed. To determine $\Phi(\eta)$, $G(\xi)$ and $\tilde{G}(\xi)$ separate computations in the three different zones of the parameter space (H,T) are required. They are schematically shown in Figure 4.1.

Fitting is performed along the zones, so the width of the three strips is taken in such a way as to minimise errors coming from the numerical differentiation. The length is mostly dictated by minimising the total error; with main sources of errors being the higher Aharony-Fisher and regular part coefficients further away from the critical point, and method imprecision closer to the critical point.

While it is possible to estimate the errors and use the estimate as a guide for the edges of the calculation zones, we mostly used a "trial-and-error" way of choosing the best parameter ranges.

For zones near $H = 0$ (zones 1 and 3 in the Figure 4.1) only M is calculated

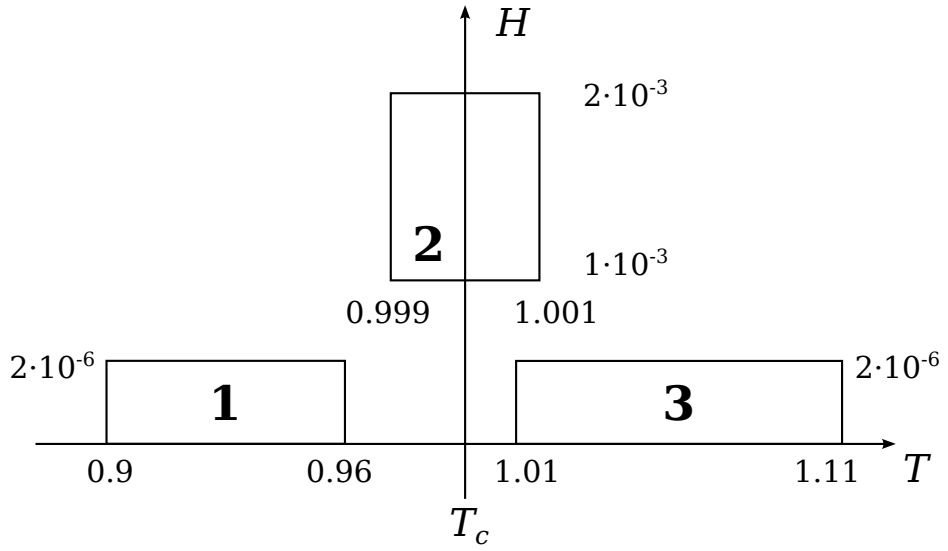

Figure 4.1: Main zones of calculations in parameter space.

(for both $T > T_c$ and $T < T_c$). For the zone near the critical isotherm (zone 2) only the internal energy is calculated, plus the free energy on the isotherm itself for the Φ_0 and regular part coefficient fits.

For all three zones computations were made in the following manner: first, for a small size of CTM states are computed sequentially (i.e., using the final state in an adjacent point as the initial state of iteration) across the zone. As a rule "heating" has slightly faster convergence than "cooling", i.e., it is better to choose the state with minimal internal energy and then increase it. The initial state for the first point (i.e., when there is no other point to take the initial state from) turned out to be not very relevant for the Ising Model. The most relevant choice would be to generate finite A and F in a natural (i.e. lattice) basis, and then to diagonalise A and convert F to its basis. In practice, however, a "fully frozen" state was always used as a seed for the smallest N. Also it should be noted that in general a solution from the adjacent point usually provides a better initial state than the exact finite CTM.

After a certain number of iterations with the smallest CTM size $N = 4$, we increase N by one (or a small natural number) and repeat the same number of iterations until the size of the CTM becomes $N = 128$. For N larger than 64 sometimes the number of iterations was divided by a small factor.

For the largest N the last iterations were performed with QUAD precision. Time consumed by this last stage was the main limiting factor for the size of CTM N, since QUAD arithmetical operations are relatively slow.

For $N = 128$ a degeneracy degree of the lowest levels in the CTM spectrum becomes quite high and further linear increase of N doesn't improve results very much even if we compute the exact CTM. The rate of decrease of eigenvalues against their number is considered in [68], the rate can be slower than exponential, and if it becomes too slow within the CTM size, then the method becomes less accurate and usable.

The actual precision for the magnetisation value in the worst cases was about 10^{-25}, and for most points comparable with the QUAD precision, that is, less than 10^{-33}. As expected from the theoretical basis, precision of the free energy is always better; the internal energy usually also has better precision than the magnetisation.

The choice of $N = 4$ as a smallest N also has some rationale behind it. $N = 1$ corresponds to the usual Kramers-Wannier approximation, $N = 2$ still could be solved analytically and also gives consistent results. $N = 3$ was tested and it turned out that sometimes it has computational artefacts and convergence oscillations with "frozen" initial condition, even for parameters far away from the critical point. This seems justifiable since that size of CTM leads to the truncation of the very first level with degeneracy, introducing large errors into the CTM spectrum. So $N = 4$ was chosen as the smallest usable size of CTM.

The choice of maximal size that is a power of 2 has a much higher chance to align well into the processor caches, hence somewhat improving performance, although no special investigations were made in that direction.

There were some additional calculations with N up to 512, but that was done mostly for test purposes and there are no direct results from the larger size of CTMs.

As a "byproduct" of making nice graphics of the scaling function, a lot of data points around the critical point were computed. We could simply draw a 3D graphic of the magnetisation vs. field and temperature, and obtain the plot in Figure 4.2. Except for the close vicinity of the critical point, the graphic is quantitatively accurate, i.e., errors from the computational procedure are well below printing resolution.

For the free and internal energies such graphics are not very exciting because the singularity is hidden behind the regular terms. The situation changes for the specific heat, where the very important logarithmic singularity is dominant and is clearly visible on the plot (Figure 4.3). Please note that the "ridge" along $H = 0$ and $T > T_c$ is not an artefact, but a natural effect connected to a different qualitative behaviour of the model in different phases.

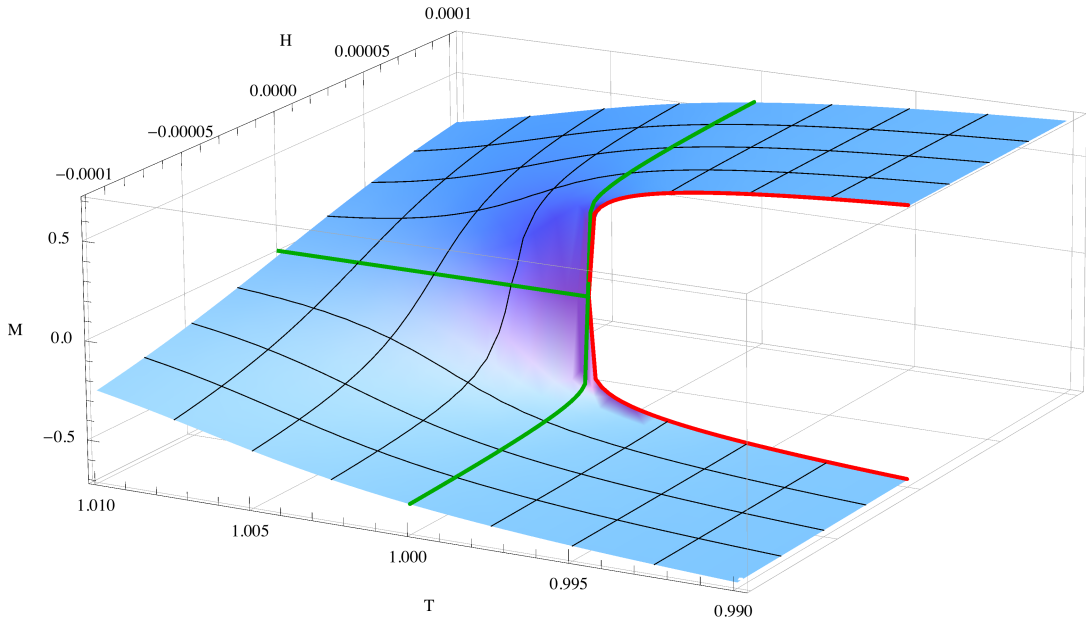

Figure 4.2: Magnetisation of the square lattice Ising model.

All technical details concerning the implementation and calculations using it are described in Appendix A. Most of the data for all computations will be available on the website [69].

Numerical Differentiation. Since only the first derivatives of the free energy could be obtained directly from the CTMRG algorithm, to employ higher derivatives in our fitting procedure one needs to make computations of the first derivative in several adjacent points and then use numerical differentiation to obtain higher derivatives of the free energy. Since our aim is to obtain as precise results as possible, very high order numerical differentiation was used. The number of points used was essentially independent of the order of the derivatives, thus the error term for higher derivatives has lower degree. Since only the values of M for $H \geq 0$ could be easily computed, a "one-sided" derivative was used; the order of the error term $O(h^n)$ is just less by one than for the symmetrical derivative with the same number of points.

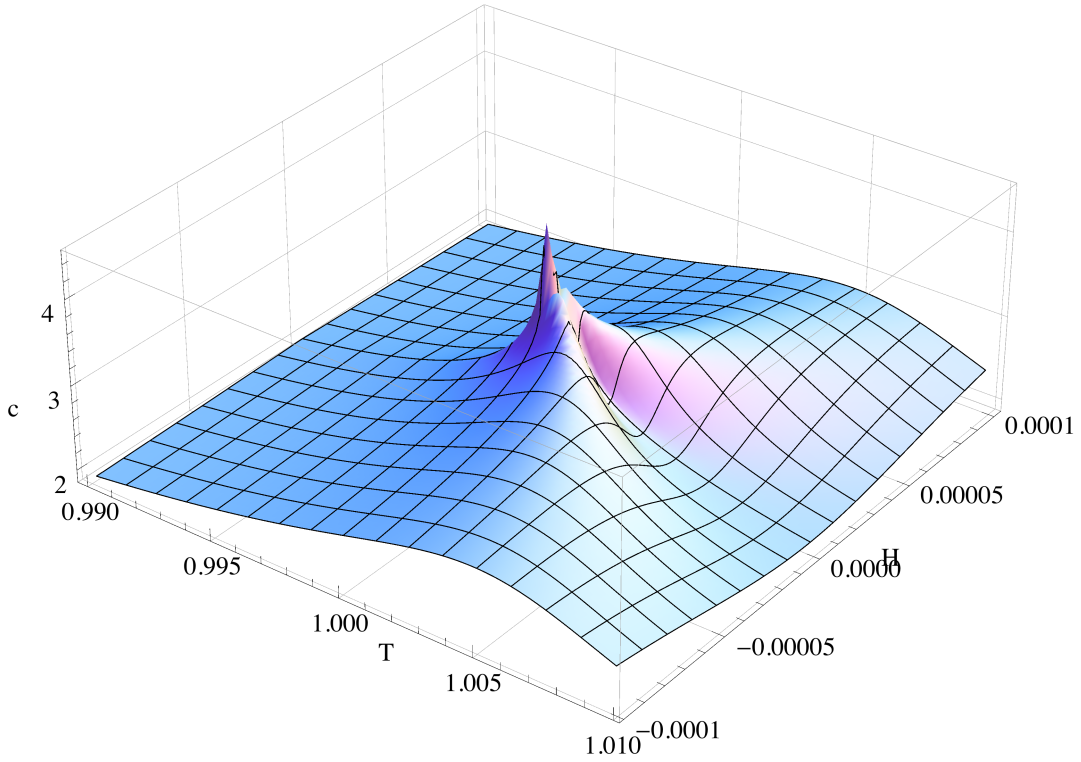

Figure 4.3: Specific heat of the square lattice Ising model.

The formula for the coefficients of the numerical derivative is [70, 71]

$$D_n[f(x_0)] = \frac{1}{h^n} \sum_{k=-s}^{m-s} a_k f(x_0 + kh) \tag{4.1}$$

where the coefficients a_k are defined by

$$\sum_{k=-s}^{m-s} a_k \cdot x^{k+s} = \sum_{j=0}^{m} \frac{(x-1)^j}{j!} \frac{d^j}{dy^j} y^s (\log y)^n \bigg|_{y=1} . \tag{4.2}$$

Here m is the number of points used for differentiation (in addition to x_0) and s is the number of points smaller than x_0.

Numerical differentiation was used directly for the M and U values. As one of the numerous tests, comparison of the directly obtained M and $\frac{\partial F(H,T)}{\partial H}$ was done not only for $H = 0$, but for many points where the model is not integrable, as well as for higher derivatives. For all points tested the difference between the values was comparable to the precision of M, so the numerical differentiation scheme is fully compatible with CTMRG, namely the algorithm seems not to introduce

any "noise" that could show up in higher derivatives.

It is possible to choose a formula with a non-uniform distribution of points to minimise not only the order of error, but also the contribution of numerical arithmetic error which could be quite large for the higher order derivatives where large coefficients in the derivative formula are mixed with small ones. But this method is rarely used, so it was decided to keep the distance between points uniform.

The last note to add about the numerical differentiation step is that sometimes an extra $1/T^n$ factor was removed after the differentiation. The factor appears if we use H/T instead of just H as in (1.3). The term was always changed just to H for computations in zone 2, since after differentiating by T the factor $1/T$ could not be excluded simply by multiplying the results and it significantly hampers fitting by making various IFT-related series much more complicated.

4.1.1 Convergence speed and precision

Most of the computation time was spent in diagonalising the extended CTM, so the speed of the diagonalising routine for the given size of CTM was the limiting factor for the computations made.

For the square and triangular lattices (where essentially only one usable algorithm was considered) no extensive tests of convergence speed were performed. Rather, most tests were designed to investigate precision and correctness of the algorithm.

For the square lattice some comparisons of different types of the lattice "inscription" were made, but mostly for correctness and not for speed of convergence. Three types of weight were compared: the usual "square" (1.23), cross-type with vertex in the centre (2.23) and cross-type without vertex in the centre (2.24). The latter variant corresponds to the two independent lattices and was studied analytically by Baxter [27]. Rough comparison tests were done using the magnetisation as a measure for all three weights. While the final precision was the same for all three weights, the convergence speed for (2.24) was much slower than for (1.23) and (2.23). Just as expected for two independent lattices the correlation function for two adjacent vertices (in the directions parallel to the edges of the Boltzmann weight) was equal to zero for any choice of parameters. While weight (2.23) showed no significant difference in performance and precision from (1.23), it was used mostly as a test for the weight with an "internal" spin summed over.

For the triangular lattice no significant difference in precision and convergence

speed between weights (2.19) and (2.20) was found, and the weight (2.19) was used in all calculations. Both weights violate the original symmetry requirements (2.12), but naturally it was found that the requirements are sometimes too strict.

The most thorough comparison of different algorithms was made for the honeycomb lattice. All three algorithms were described in Section 2.2.3. The "auxiliary-matrix" algorithm showed the worst convergence speed and no correct results were ever obtained for $H \neq 0$. The main competition was between the "big-block" and "asymmetric-CTM" algorithms. The "big-block" showed really fast convergence in terms of the number of iterations, which was expected since at each step the lattice is extended with four weights instead of one. But the real-time speed is obviously hampered by the large size of the extended CTM. On the other hand, the "asymmetric-CTM" algorithm is 2-3 times faster per iteration than "big-block", allowing for faster convergence. Counting by the real-time "asymmetric-CTM" was faster (but not more than by 30% margin) than the "big-block" algorithm, so the "asymmetric-CTM" was used for all bulk computations. The last but not the least observation is that if the size of the CTM (i.e. not arithmetic precision) is the limiting factor for the magnetisation precision, then even after the iteration convergence has completed, the magnetisation precision in the "big-block" algorithm is one or two orders better than for the "asymmetric-CTM". Since the accuracy for most points is limited by arithmetic precision, this doesn't affect the choice of algorithms.

Several numerical effects independent of the lattice type are also worth noting. The first effect is random spin "flips", i.e. spontaneous change of the of M. They happened only when $H = 0$ and only for QUAD precision. The probability of the flips is not entirely clear, for several dozens of datapoints after several hundreds iterations (for the largest N only, since no flips happen for the normal precision irrespective of the number of iterations) up to ten points had the wrong sign of M. It is interesting that for the small enough values of H below zero (and sufficiently away from the critical point to prevent immediate change of the sign of H), no such flips occur for any precision, i.e., the effect is strictly confined to $H = 0$.

For $H < 0$ another effect takes place. Since the initial state for all points in the dataset (except for the first one) is taken from the previous point, points for $H < 0$ and $T < T_c$ would have initial state with $M \geq 0$. It turns out that despite the large number of iterations, the states don't revert to $M < 0$ immediately, and for a given number of iterations, the further away we are from the critical point the stronger field $H < 0$ is required to make M change sign. The transition field

also depends on the matrix size N, and the effect is noticeable for all N under consideration. We didn't research the effect much, as the study is complicated due to the rather random transition from $M > 0$ to $M < 0$ near the critical value of H. The effect is certainly of considerable interest because the derivatives of M obtained using values from the $H < 0$ side have correct values, i.e. values identical to the one-sided derivatives.

4.1.2 CTM structure

The spectrum of the CTM is one of the core reasons for the CTMRG success. It is worth showing the actual spectrum of the truncated CTM for various (H, T) values and compare it (at least qualitatively) with the exact solution and other theoretical considerations.

Apart from the variational methods, CTMs were used in the analytical solution of the 2D Ising model [27] without any reference to the variational approach or renormalisation group, so the main thing to compare with theory is the fixed point of CTMRG iterations. An explicit spectrum was derived for the infinite CTM in the diagonal representation, which can be used very naturally to derive (1.17a). The spectrum has a rather simple structure, shown in Figure 4.4 for the Ising model with weight (2.24). The whole spectrum consists of equidistant (in a logarithmic scale) levels, the degree of degeneracy of each level equals the number of partitions of the number of the level [72]. The "distance" between levels equals

$$x = \frac{a_{i+1}}{a_i} = \exp\left[-\frac{\pi K'(k)}{4K(k)}\right] = q^{1/4}, \tag{4.3}$$

where K and K' are compete elliptic integrals of the first kind and its complementary, q is the elliptic nome.

For the most used weight (1.23) the spectrum has a slightly different form: x is the same and the degeneracy of each level equals the number of odd partitions of the level's number, or, equally, the number of distinct partitions [73].

The presence of degeneracy at each level could be seen as indirect evidence of model integrability.

Due to truncation, the CTM spectrum differs from its exact form even for $H = 0$; in the second half of the spectrum some eigenvalues are out of their levels. For the weight (1.23) in zero field, a typical CTM spectrum is shown in Figure 4.5.

For non-zero field the 2D Ising Model is non-integrable, and degeneracy of

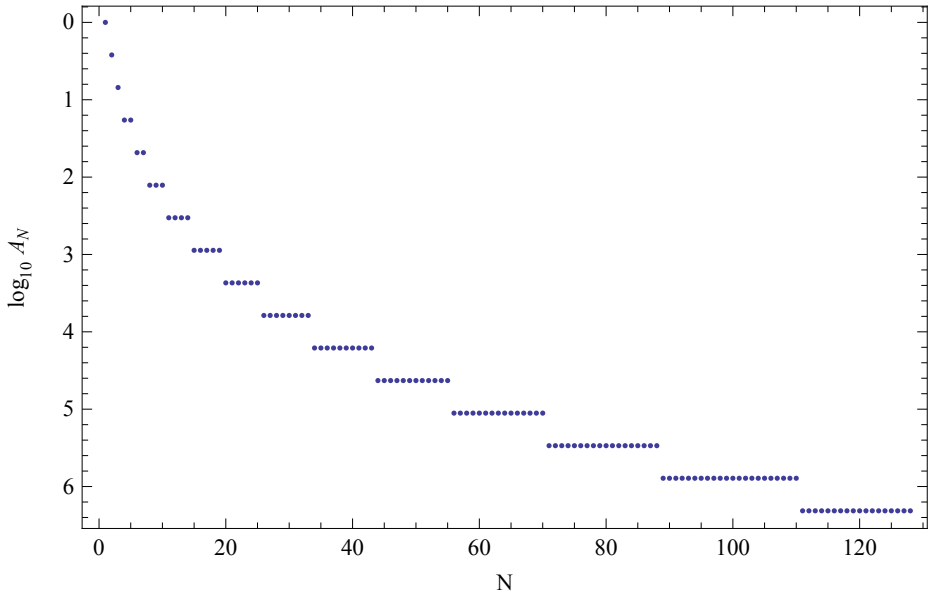

Figure 4.4: The spectrum of the CTM for the 2D Ising Model in zero field, $T = 1.02$.

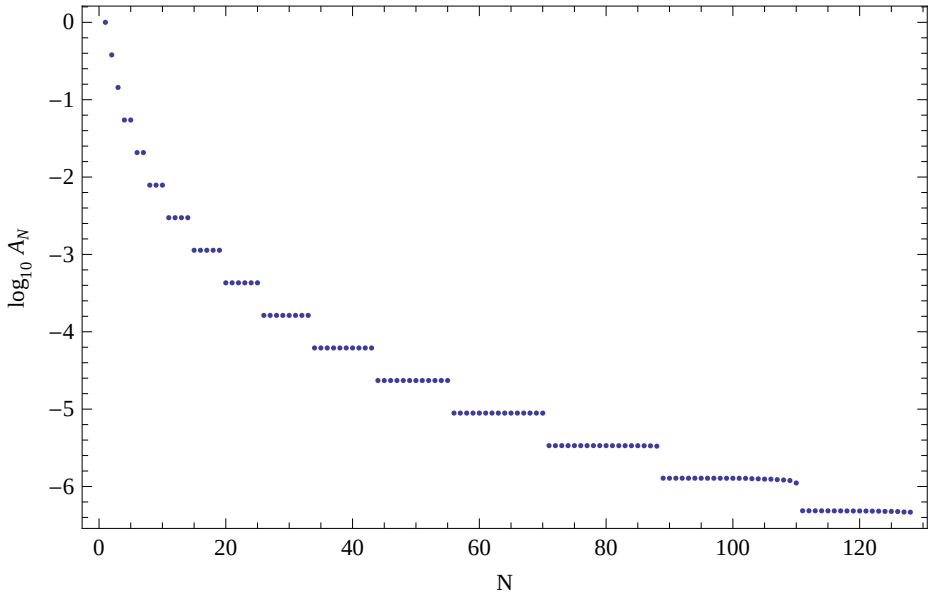

Figure 4.5: The spectrum of the CTM for the 2D Ising Model in zero field, from the real calculations, $T = 1.02$.

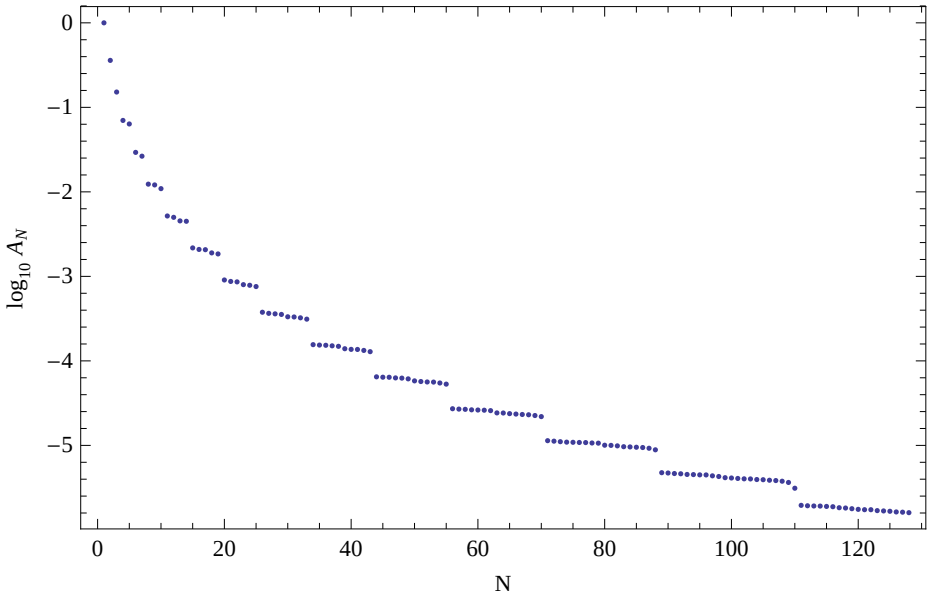

Figure 4.6: The spectrum of the CTM for the 2D Ising Model in a weak field, $T = 1.01$, $H = 0.00009$.

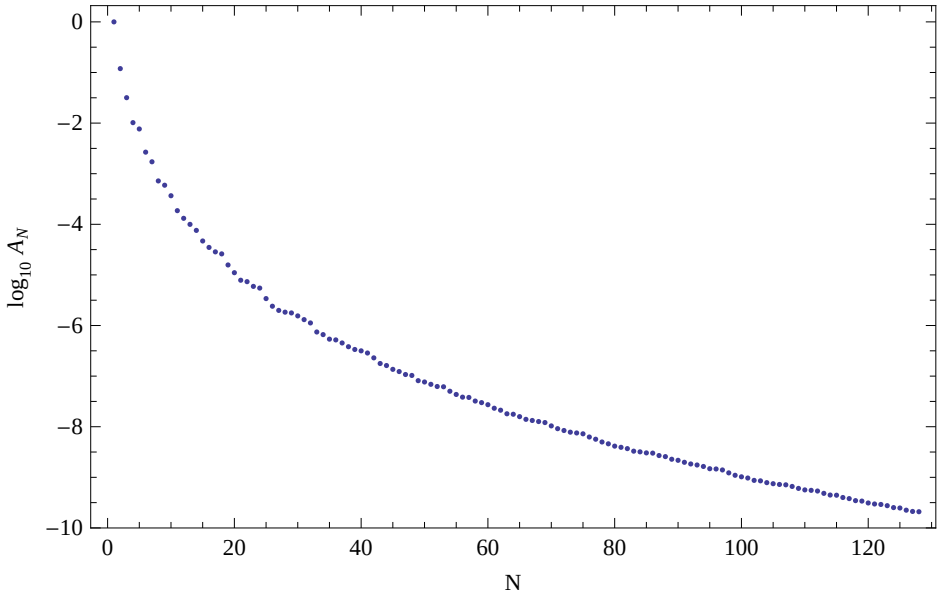

Figure 4.7: The spectrum of the CTM for the 2D Ising Model in a field for the critical temperature and field $H = 0.005$.

levels doesn't occur. In a weak magnetic field the CTM spectrum levels start to "deform" slightly, but still stay in groups. A typical spectrum is shown in Figure 4.6. It is possible that an additional careful study of such deformations may reveal some regularities in the deformation and produce a semi-analytic approximation for the solution, if the field is weak enough.

In a stronger field degeneracy disappears and all eigenvalues are arranged along the single line that represents an asymptotic behaviour which is rather general. An example is shown in Figure 4.7, and it actually roughly corresponds to another exact solution of the Ising model – Zamolodchikov's E_8 solution, mentioned in Section 3.2.

No spectral properties of other models were studied, except that eigenvalue behaviour was used as yet another of the methods to determine a critical temperature of various models. For the three-state Ising and Blume-Capel models the distance between levels x was computed as ratios of several first eigenvalues. The core of the method was to find T such that $x(T)$ is maximal, since the extreme value of x should be reached at the critical point, making the spectrum singular in the exact case. Different values of x were computed, e.g. as ratios of $A_0(i)$ and $A_1(i)$ for $i = 1, 2, 3$, or as $A_0(i)/A_0(j)$ for different i,j. The problem was that values of x obtained were slightly different, but different enough to introduce sizeable errors in the computed T_c. Another problem was that values of x keep "flowing" with iterations even after many thousands of iterations with large CTMs. For the Blume-Capel model these x values were also subject to some oscillations during the iterations, greatly diminishing their usefulness in estimating the critical point.

4.2 Fitting the expansion coefficients

As a core result of this work, the series expansion coefficients Φ_i, G_i and $\tilde{G}_i$ were determined with high precision using data obtained from the lattice computations. No previous results for the lattice Ising models with similar precision were previously known. The only complete result to compare with is the "Truncated Free-Fermion Space Approach" for the IFT presented in [17], which was already mentioned in Chapter 3.

Apart from using robust numerical algorithms, to improve the precision we used as many exact results as possible. These include information about the structure of the free energy (i.e., division into the regular and singular parts), known

coefficients in the regular part and Aharony-Fisher scaling fields, the analytic structure of the singular part and the known lower oder expansion coefficients of the scaling function. After incorporating knowledge of all these factors into the expansions of the free energy derivatives and linearly fitting the series expansions against numerical derivatives of the free energy, the coefficients of the scaling function appear directly as coefficients in the linear regression model, with the fit errors being usually larger than the numerical errors coming from the lattice computations.

At the time of computation of the triangular and honeycomb coefficients, the data similar to [32] for the square lattice were not available. It appeared later [33] when this thesis was near completion (and after [74] was published), so it is not fully incorporated into the current work.

4.2.1 Complete Series

To do the fit we need to combine all series coming from the Aharony-Fisher lattice scaling fields, the regular part of the free energy and the expansion of the scaling function to be able to determine the unknown parameters. All fits are done in H and τ variables.

Three main sources of unknown variables are the

- Regular part of the free energy (3.37).
- Aharony-Fisher scaling fields (3.43) and (3.44), described in detail in Section 3.3.2.
- Scaling function itself: (3.25), (3.27) and (3.29).

The subleading part of the free energy $F_{\text{sub.}}$ is usually small enough to be excluded from most of the fits. The only places where the subleading part is taken into account are G_2 and $\tilde{G}_2$ coefficient fits, which get special treatment, mostly with the help of [32].

After each successive differentiation the complexity of the coefficients increases, so it is not worth studying any coefficients except the leading one. Except for the first two derivatives for Φ (plus Φ itself), the leading coefficient in the fitting series contains an expansion coefficient of the scaling factor times a known exact factor.

It also should be noted that most of series below are obtained not from substituting all formulas into each other and then differentiating. This procedure was

tried and it takes too much symbolic computation time in most cases due to the coefficient complexity and Mathematica's poor handling of $O(x^n)$ notation. The more efficient approach is to have expansions for all formulas (even for values that are known exactly), substitute the series into each other, and after each symbolic differentiation manually remove unneeded coefficients that are not contributing to the coefficients we are interested in. To prevent significant parts from being omitted some insignificant "guarding" coefficients were always kept in the series as a replacement for $O(x^n)$ terms.

Φ_i coefficients series. The first set of series applies to zone 2, i.e., the critical isotherm, and is fitted against the H variable. Since there is no exact lattice solution on the critical isotherm, no part of the series extends to arbitrary higher order in the fitting variable.

We differentiate w.r.t. the T variable and not τ (which the main series are in), thus coefficients are riddled with $\left.\frac{d^n\tau(T)}{dT^n}\right|_{T=1}$ factors. In practice this doesn't create any difficulties.

The series for the free energy up to H^4 with $\tau = 0$ is

$$\begin{aligned} F(H,0) &= A_0 + C_h^{16/15}\Phi_0 H^{16/15} + B_0 H^2 + C_h^{8/15}\mu_h\Phi_1 H^{38/15} \\ &+ \frac{16}{15}C_h^{16/15}e_h\Phi_0 H^{46/15} + O(H^4). \end{aligned} \tag{4.4}$$

This gives us Φ_0 and Φ_1 plus non-universal Aharony-Fisher field coefficients e_h, μ_h and a regular free energy coefficient B_0. The non-universal coefficients and a regular part were known only for the square lattice. The first logarithm appears at H^4 order.

After one differentiation w.r.t. T we obtain the series for the internal energy, with order up to H^2,

$$U(H,0) = A_1 t_1 - C_h^{8/15}C_\tau t_1\Phi_1 H^{8/15} + \frac{16}{15}C_h^{16/15}\Phi_0 c_1 t_1 H^{16/15} + O(H^2), \tag{4.5}$$

where c_1 is a coefficient from $c(\tau)$ expansion (3.46) and

$$t_n = \left.\frac{d^n\tau(T)}{dT^n}\right|_{T=1}. \tag{4.6}$$

The first logarithm here is at H^2 order. The expansion gives Φ_0 and Φ_1 which enter it without unknown coefficients, so it serves as the most precise test of exact values of these coefficients. Even without knowledge of their exact values it is

possible to substitute fitted values to (4.4) and obtain e_h and μ_h values fairly accurately.

The series for the specific heat is the most difficult one to fit since the leading singularity in it is a logarithm and the coefficient of Φ_2 is still regular and coupled to a logarithm. Careful treatment of all scales is needed, because even physically insignificant factors in $m(H,T)$ would contribute to the term which contains the Φ_2 coefficient and make its value incorrect. Up to H^2 the series is

$$\begin{aligned} c(H,0) &= 2t_1^2 A_2 + A_1 t_2 + 2C_t^2 t_1^2 \Phi_2 + \frac{4}{15\pi} C_t^2 t_1^2 \log C_h H \\ &+ C_h^{8/15} C_t \Phi_1 \left(2a_1 t_1^2 + \frac{16}{15} c_1 t_1^2 + t_2 \right) H^{8/15} \\ &+ C_h^{16/15} \Phi_0 \left(\frac{16}{225} c_1^2 t_1^2 + \frac{32}{15} t_1^2 c_2 + \frac{16}{15} c_1 t_2 \right) H^{16/15} \\ &+ \frac{6 C_t^2 t_1^2 \mu_h \Phi_3}{C_h^{8/15}} H^{22/15} + O(H^2). \end{aligned} \tag{4.7}$$

Here a_n are coefficients of the $a(\tau)$ expansion (3.45). The only usable term to fit is the constant term with Φ_2, although other terms with known coefficients could be subtracted to improve the Φ_2 precision. Φ_3 could also be roughly estimated from the $H^{22/15}$ term, but the derivative of the specific heat provides a significantly more accurate result.

Starting from the third derivative the leading singularity is a power singularity and its coefficient contains Φ_i

$$\frac{\partial^n F(H,\tau(T))}{\partial T^n} = (t_1 C_\tau)^n \Phi_n n! (C_h H)^{\frac{8(2-n)}{15}} + O\left(H^{\frac{8(1-n)}{15}} \right). \tag{4.8}$$

No additional notes about this expansion follow, except as already mentioned, the lower the order, the more "dummy" terms are recommended to be added into the fit. Although it is perhaps possible to get some information, coefficients beyond the leading order are very cumbersome and unusable for fitting.

G_i and $\tilde{G}_i$ coefficients series. Series for the G coefficients are considered in the same way as others, except for the fact that for G_2 and $\tilde{G}_2$ we tried to include some traces of $F_{\text{sub.}}$ into the fit.

The susceptibility series used for G_2 and $\tilde{G}_2$ fitting is

$$\chi(\tau) = -\frac{2C_h^2}{(\sqrt{2}|\tau|)^{7/4}} F_{\text{AF}}(\tau) + B_\chi(\tau), \tag{4.9}$$

where $F_{\rm AF}(\tau) = a(\tau)^{-\frac{7}{4}} c(\tau)^2$ and $B_\chi(\tau)$ is from (1.21). After expanding known series $a(\tau)$ and $c(\tau)$, the series in the first term takes the form

$$F_{\rm AF}^{(s)}(\tau) = 1 + \frac{\tau}{2} + \frac{5\tau^2}{8} + \frac{3\tau^3}{16} - \frac{11\tau^4}{192} - \frac{17\tau^5}{384} + \frac{97\tau^6}{3072} + O(\tau^7), \tag{4.10a}$$

$$F_{\rm AF}^{(t)}(\tau) = F_{\rm AF}^{(h)}(\tau) = 1 + \frac{\tau}{3} + \frac{59\tau^2}{288} + \frac{13\tau^3}{288} + \frac{65\tau^4}{165888} - \frac{967\tau^5}{497664} + \frac{9721\tau^6}{47775744} + O(\tau^7) \tag{4.10b}$$

In [32, 33] it is shown that the actual $F_\chi(\tau) \neq F_{\rm AF}(\tau)$, and the difference is due to contribution of irrelevant contribution. It also turns out that the $\chi(\tau)$ expansion is good for estimating contribution of $F_{\rm sub.}$ quantitatively. The basic analysis presented in Section 3.3.3 tells us that for the square lattice the discrepancy between $F_{\rm AF}$ and F_χ begins at order τ^4, and for the triangular and honeycomb lattice at order τ^6.

Series that were used for determination of $\tilde{G}_3$ contained at least one non-dummy higher-order coefficient

$$\left.\frac{\partial^3 F(H,T)}{\partial H^3}\right|_{H=0} = -\frac{6\tilde{G}_3 C_h^3}{(-C_t\tau)^{\frac{29}{8}}}(1+\alpha_1\tau) + O\left(\tau^{-\frac{21}{8}}\right), \tag{4.11}$$

where $\alpha_1^{(s)} = -\frac{3}{4}$, $\alpha_1^{(t)} = \frac{2}{3}$ and $\alpha_1^{(h)} = \frac{1}{2}$. These coefficients were found numerically to very high precision (with error less than 10^{-5}).

For $n \geq 3$ all series take form

$$\left.\frac{\partial^n F(H,T)}{\partial H^n}\right|_{H=0} = \frac{G_n n! C_h^n}{\left[(C_t\tau)^2\right]^{\frac{15n}{16}-1}} + O\left(\tau^{3-\frac{15n}{8}}\right). \tag{4.12}$$

4.2.2 Regression procedure

After numerical differentiation and removal of trivial factors, data points from zones 1-3 are fitted against the series presented in the last section. There are many problems that could occur at this stage. Errors in the data points are different for each point, so proper choices of data points for each series is required. The fit could be too rough if not enough terms are included. It could become destabilised if too many terms (beyond the number needed to achieve maximal possible precision) are included. Error estimates from the fit routine (Regressing Function) could be underestimated or just random, but obviously results could not be better than the precision from the original numerical data. So it is man-

ual work to tweak all these parameters in order to minimise errors and provide plausibility for the expansion coefficient values.

Regression Function. For all standard linear regressing a built-in Mathematica function `LinearModelFit[]` was used [75]. It has rather conventional inputs: array of variable and function value pairs, and a list of terms to fit against. It outputs the `ModelFit` object, and we are mostly interested in its `ParameterTable` component, which is a table. Rows of the table corresponds to the terms (term 1 is implicitly included, unless a special option is passed to the function), columns are term, coefficient estimate, "Standard Error", "t Statistics" and "P-Value". While not all prerequisites hold in our data to use "Standard Error" and "P-Value" as a rigorous error estimate, we mostly quote the error given in the "Standard Error" column or variation arising from inclusion or exclusion of additional terms and data points to the fit, whichever is greater.

To visually estimate the error of the fit, the `FitResiduals` component of the `ModelFit` object could be used. One could use this array and simply plot it. If the graphic looks rather monotonous, and not like a sinusoid, then it is likely that there is some significant error in one single term of the fit, and not an error in the "tail" of terms that are not included. Of course this could not serve as a proper error estimate, but allows one to quickly judge the quality of fit while choosing various parameters to improve precision.

In the paragraphs below we list several observations and peculiarities of the coefficient fitting. We don't list very specific information, like which point numbers were used for the fit, all high-order "dumb" terms etc, since our choice may not be the very best and it is not dictated by any constructive requirements.

Low-order Φ_0, Φ_1, Φ_2 coefficients. Apart from the coefficients shown in (4.4), the following terms were added as "dumb" ones: x^4, x^6, and for (4.5): x^2, x^4, $x \log x$. As for most of the Φ_i fits, all coefficients were used, as dropping fringe data points doesn't give any significant improvements.

Since Φ_0 and Φ_1 are known exactly, the main purpose for the fit was overall testing and obtaining the values of the non-universal coefficients μ_h and e_h. The latter was done with substitution of the exact Φ_0 and Φ_1 values into the series (4.4) and (4.7) and then fitting unknown coefficients μ_h and e_h, as well as B_0. For the square lattice, the B_0 coefficient was known and thus was subtracted from the data before fitting the (4.4) series in addition to substituting exact Φ_0 and Φ_1 values. This didn't give any decisive improvement for e_h and μ_h though.

For Φ_2, after testing that the leading logarithm term has indeed the predicted value, it was subtracted from the data to get the constant term containing Φ_2. This gave around three extra digits for the Φ_2 value.

Higher-order Φ_i coefficients. Obtaining other higher coefficients is rather straight forward from the leading term of (4.8). Not many dumb terms (except for Φ_3) were used, since for the higher order coefficients it becomes difficult to obtain their order in H, let alone any estimation of the coefficient values themselves. Usual degrees of "dummy"' terms included $\frac{8}{15}$, $\frac{16}{15}$, $\frac{22}{15}$, $\frac{32}{15}$ (relative to the leading term) plus integer degrees. For the highest Φ_i, some points were dropped from the fit to reduce the number of degrees of freedom in case not enough dummy terms are included. In contrast with the other fits of G_i, the range of H was chosen in such a way that it wasn't important which coefficients were dropped, as only their number matters, and not the distance to the critical point.

Absolute values of Φ_i decrease with i. This is a natural consequence of the $\Phi(\eta)$ series being convergent inside a circle with radius $\eta_c > 1$. Since only the leading term is used, it is possible to get many more coefficients with large i; one needs only to compute more data points in (H, T)-space to have enough points to use for numerical differentiation.

G_2, $\tilde{G}_2$ from the susceptibility series. For these fits fine tuning of which data-points and which terms to include was done more than for any other fit. For $T < T_c$ at least 10 data points at both ends of the T range were dropped, so the fit essentially ranged from $T = 0.91$ to $T = 0.95$. Whether to exclude additional points was strongly dependent on the number of terms being fitted, but for some optimal number of terms included into $F_\chi(\tau)$ and $B_\chi(\tau)$ series it was possible to obtain a good fit without dropping any additional data points.

For higher temperature the fit was even more unstable. The basic usable range is $T = 1.03 - 1.10$.

Coefficients in $F_\chi(\tau)$ were fit successively. After the fit of the τ^n coefficient and verifying its value against (4.10), the term was substituted back into (4.9) and the fit was redone for the τ^{n+1} coefficient. Terms up to τ^3–τ^4 could be fitted without any knowledge of $B_\chi(\tau)$.

For the square lattice we took $B_\chi(\tau)$ and subtracted its value from $\chi(\tau)$, enabling the fit to get a couple more terms in $F_\chi(\tau)$. For the triangular and honeycomb lattices the constant term from $B_\chi(\tau)$ provides an independent check of the B_0 value, although the value from (4.4) is usually more accurate.

Higher-order G_i coefficients, As for Φ_i, these fits reveal no peculiarities. The core difference is that the series for $\tilde{G}(\xi)$ is only asymptotically convergent, thus the series coefficients may not decrease with order, making fits a bit more troublesome. In comparison to the Φ_i coefficients, where additional derivatives could give higher coefficients using the same fit procedures, obtaining higher $\tilde{G}_i$ coefficients would require rethinking the entire scheme to get rid of very large numbers at the input of the fit. Also the higher $\tilde{G}_i$ are of limited use because of the absence of absolute convergence.

As for the higher Φ_i, no subleading coefficients are analysed, although the coefficients in some cases might be simpler than for Φ_i. At least for $\tilde{G}_3$ and $\tilde{G}_4$ there were attempts to get information from the fitting of subleading terms. The terms contain $\tilde{G}_2$, but the achieved precision is well below the precision from the fits specific for these coefficients. It was possible to subtract these terms using their semi-exact value, but no significant gain in precision of $\tilde{G}_3$ or $\tilde{G}_4$ resulted.

4.2.3 Non-universal coefficients

Here we summarise our knowledge of the non-universal coefficients that were (or in principle could be) obtained from the fits. Table 4.1 shows these coefficients.

Square lattice. Due to more research being done for the square lattice than for the other types, almost all non-universal coefficients were already known. Confirming their values from the direct lattice calculations is important for overall consistency.

Triangular and Honeycomb lattices. For the triangular and honeycomb lattices the coefficients in the table were not known in advance, so they were originally obtained from the Φ_0 and Φ_1 fits and used later in the $b(\tau)$ and $d(\tau)$ functions.

For the honeycomb lattice the situation is pretty much the same as for the triangular one, except the results obtained were finally verified at the same time as [33] was published, i.e., their results were not used directly, but might have been achieved earlier than our results.

For these coefficients errors are very difficult to estimate, so "in the worst case" one should assume precision around ± 5 in the last digit.

Coefficient	Square	Triangular	Honeycomb
B_0	0.0520666225469	0.0247805582(2)	0.110763138(1)
μ_h	0.071868670814	−0.010475(1)	0.3040413(2)
e_h	−0.00728(30)	0.00129(1)	−0.04348(1)

Table 4.1: Lattice-dependent coefficients obtained from the fits (or from references if available)

4.2.4 Irrelevant Operators

The contribution of the irrelevant operators was estimated via the susceptibility fit, following [32] and [19].

For the square lattice after subtracting $B_\chi(\tau)$ we found that

$$F_{\mathrm{AF}}^{(s)} - F_\chi^{(s)} = -\frac{1}{384}\tau^4 + O(\tau^5), \tag{4.13}$$

in full accordance with [32]. The relative precision of τ^4 term coefficient was around 0.02% .

For the triangular lattice results were not known at the moment, so determination of the contribution from irrelevant operators was more troublesome. Analysis in [19] tells us that the first deviation of $F_\chi(\tau)$ from $F_{\mathrm{AF}}(\tau)$ happens at τ^6 which is beyond our precision. Using the fit with unknown $B_\chi(\tau)$ we were able to show that

$$F_{\mathrm{AF}}^{(t)} - F_\chi^{(t)} = O(\tau^5), \tag{4.14}$$

although the precision in order τ^4 was comparable to or greater than $\frac{1}{165888}$ (the denominator here is the denominator of the τ^4 coefficient from $F_{\mathrm{AF}}(\tau)$, formula (4.10b)), so the result is not very decisive.

We didn't undertake any attempt to find traces of the irrelevant operators on the honeycomb lattice; the analysis mentioned in Section 3.3.3 shows that deviation should occur at the τ^6 coefficient, i.e., in the same order as for the triangular lattice.

4.3 The Scaling Function

As the main "visual" result of the thesis, a high-quality plot of the scaling function has been obtained. It is shown in Figure 4.8. The point-wise data that were used for the plot could be employed for fast calculation of $\Phi(\eta)$ via interpolation, especially in the areas where series convergence is the worst, e.g., when $|\eta| \approx 2.43$.

The data for the plot were obtained separately from the computations of the scaling function expansion coefficients, i.e., not using any points from Zones 1-3. There were three main sets of double precision calculations relatively close to the critical point, with the same range of $T = 0.99 - 1.01$ and three different scales of field: 10^{-3}, 10^{-4} and 10^{-5}. Each set contained 10000 points arranged in a grid of 100×100 points. The same sets were used to make Figures 4.2 and 4.3. Typical CTM size was around 64 since the proximity of the critical point required many more iterations than usual. Some points very close to criticality were excluded to account for poorer precision of the method very close to the critical point, around 7000 points were eventually used from the set with $|H| < 10^{-4}$. The "thickness" of the curve in the plot served as a measure of appropriate point choice, it varies from 10^{-9} to 10^{-6}, but for any value of η it is well beyond the possible resolution of the plot at the presented scale.

All three series $\Phi(\eta)$, $G_{\text{low.}}(\xi)$ and $G_{\text{high}}(\xi)$ are also shown, as well as their domains of convergence.

In Figure 4.9 the plot of the regular part $\tilde{\Phi}(\eta)$ of the scaling function is shown. For $|\eta| < 2.43$ $\tilde{\Phi}(\eta)$ is strongly dominated by the η^2 term.

In [17] a table with values $\Phi(\eta)$ is given for $\eta = -5, -4, \ldots 4, 5$. As an additional test (and by a request of one of the [17] authors) we computed the same values on the square lattice without using interpolation, by directly choosing T and H to match integer η. Almost the same numbers were obtained, and they are shown in Table 4.4.

At the end, an interesting question arises; namely, which of results presented in this chapter could be obtained without help of the exact solutions. Extreme cases of that situation are presented in Chapter 5 where either nothing is known about the model exactly, even the critical temperature (as for the spin-1 Ising model), or only critical properties are know (as for the Potts model). Consequently, accuracy of the numerical fits diminished greatly. Slightly more interesting is a case when we have some exact results (but less than for the Ising model), and try to estimate the rest using numerical computations. During numerous tests of the computation algorithms and fitting procedures we tried to obtain as much as possible from the numerical data before resorting to the exact solution, and found out the following.

While we realised that it is possible to compute the scaling function without using any extra data (or without using, for example, "semi-numerical" results for susceptibility), the numerical precision needed to carry out that programme should be much better than we can currently afford, the biggest obstacle being

improving precision of the raw data (i.e. entries of the CTMs). QUAD precision is the most precise number format that can be used efficiently (at least, on current computers), and even doubling that precision would require enormously large amount (in comparison to the amount we used) of computation resources. The argument could be especially clearly seen for the specific heat series fit (4.7) which is notoriously difficult to control whilst it has an exactly-known coefficient at the logarithmic order.

As our main aim was to achieve new results or confirm pre-existing results using independent methods, we decided to sacrifice self-sufficiency of our fitting procedures for the sake of obtaining as precise Φ_i as possible. Without such a trade off, the precision of our results would be much worse; although the results could be possibly achieved without the trade off provided unlimited computational resources.

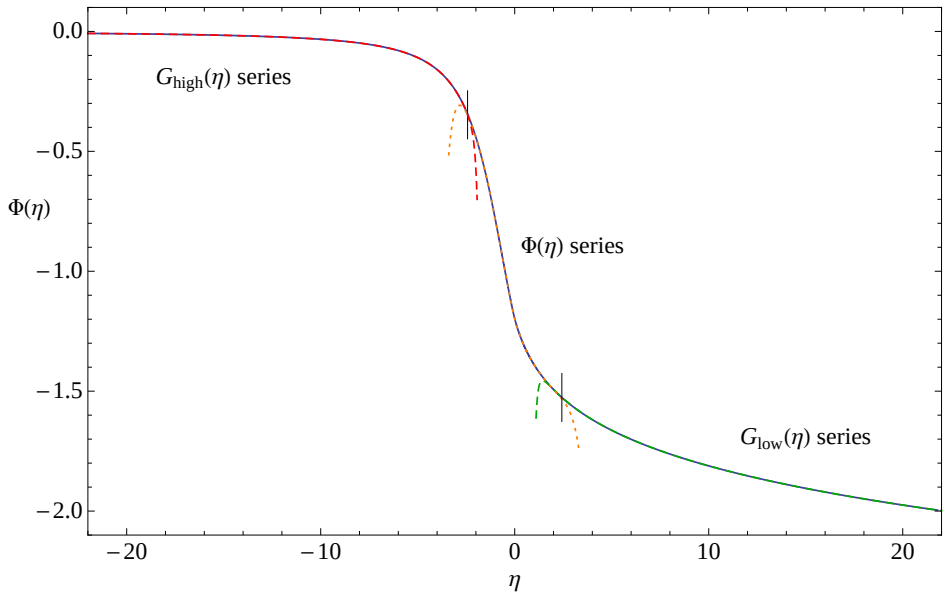

Figure 4.8: Graphic of the scaling function.

	Tr. lat. CTM	Sq. lat. CTM	Hc. lat. CTM
$\tilde{G}_1$	$-1.357838341706595(2)$	$-1.3578383417066(1)$	$-1.357838341706594(3)$
$\tilde{G}_2$	$-0.048953289720(2)$	$-0.048953289720(1)$	$-0.048953289720(2)$
$\tilde{G}_3$	$0.0388639290(1)$	$0.038863932(3)$	$0.0388639289(1)$
$\tilde{G}_4$	$-0.068362121(1)$	$-0.068362119(2)$	$-0.068362120(2)$
$\tilde{G}_5$	$0.18388371(1)$	$0.18388370(1)$	$0.18388372(2)$
$\tilde{G}_6$	$-0.659170(1)$	$-0.6591714(1)$	$-0.659169(2)$
$\tilde{G}_7$	$2.93763(2)$	$2.937665(3)$	$2.93765(1)$
$\tilde{G}_8$	$-15.57(2)$	$-15.61(1)$	$-15.60(1)$
G_2	$-1.84522807823(1)$	$-1.8452280782328(2)$	$-1.845228078233(1)$
G_4	$8.3337117508(1)$	$8.333711750(5)$	$8.3337117508(1)$
G_6	$-95.16897(3)$	$-95.16896(1)$	$-95.1690(1)$
G_8	$1457.8(2)$	$1457.62(3)$	$1457.9(2)$
G_{10}	$-25891(2)$	-25889	$-25888(1)$
Φ_0	$-1.197733383797993(1)$	$-1.197733383797993(1)$	$-1.197733383797992(1)$
Φ_1	$-0.3188101248906(1)$	$-0.318810124891(1)$	$-0.3188101248905(2)$
Φ_2	$0.1108861966832(3)$	$0.110886196683(2)$	$0.1108861966829(3)$
Φ_3	$0.01642689465(1)$	$0.01642689465(2)$	$0.01642689466(2)$
Φ_4	$-2.6399783(1) \times 10^{-4}$	$-2.639978(1) \times 10^{-4}$	$-2.6399784(1) \times 10^{-4}$
Φ_5	$-5.140526(1) \times 10^{-4}$	$-5.140526(1) \times 10^{-4}$	$-5.140526(2) \times 10^{-4}$
Φ_6	$2.08866(1) \times 10^{-4}$	$2.08865(1) \times 10^{-4}$	$2.08865(1) \times 10^{-4}$
Φ_7	$-4.481969(2) \times 10^{-5}$	$-4.4819(1) \times 10^{-5}$	$-4.48197(1) \times 10^{-5}$
Φ_8	$3.194(1) \times 10^{-7}$	—	$3.192(2) \times 10^{-7}$
Φ_9	$4.313(1) \times 10^{-6}$	—	$4.312(1) \times 10^{-6}$
Φ_{10}	$-1.987(2) \times 10^{-6}$	—	$-1.989(2) \times 10^{-6}$
Φ_{11}	$4.32(1) \times 10^{-7}$	—	$4.35(2) \times 10^{-7}$

Table 4.2: Scaling function expansion coefficients on the different lattices.

	Tr. lat. CTM	IFT [17]	References	
$\tilde{G}_1$	$-1.357838341706595(2)$	-1.35783835	$-1.357838341706595496...$	[59]
$\tilde{G}_2$	$-0.048953289720(2)$	-0.0489589	$-0.0489532897203...$	[31, 76, 32]
$\tilde{G}_3$	$0.0388639290(1)$	0.0388954	0.0387529 [77]; 0.03893	[78]
$\tilde{G}_4$	$-0.068362121(1)$	-0.0685060	-0.0685535 [77]; $-0.0685(2)$	[79]
$\tilde{G}_5$	$0.18388371(1)$	0.18453	—	
$\tilde{G}_6$	$-0.659170(1)$	-0.66215	—	
$\tilde{G}_7$	$2.93763(2)$	2.952	—	
$\tilde{G}_8$	$-15.57(2)$	-15.69	—	
G_2	$-1.84522807823(1)$	-1.8452283	$-1.845228078232838...$	[31, 76, 32]
G_4	$8.3337117508(1)$	8.33410	$8.33370(1)$	[63]
G_6	$-95.16897(3)$	-95.1884	$-95.1689(4)$	[63]
G_8	$1457.8(2)$	1458.21	$1457.55(11)$	[63]
G_{10}	$-25891(2)$	-25889	$-25884(13)$	[63]
Φ_0	$-1.197733383797993(1)$	-1.1977320	$-1.19773338379799339...$	[60]
Φ_1	$-0.3188101248906(1)$	-0.3188192	$-0.31881012489061...$	[80, 61]
Φ_2	$0.1108861966832(3)$	0.1108915	—	
Φ_3	$0.01642689465(1)$	0.0164252	—	
Φ_4	$-2.6399783(1) \times 10^{-4}$	-2.64×10^{-4}	—	
Φ_5	$-5.140526(1) \times 10^{-4}$	-5.14×10^{-4}	—	
Φ_6	$2.08866(1) \times 10^{-4}$	2.09×10^{-4}	—	
Φ_7	$-4.481969(2) \times 10^{-5}$	-4.48×10^{-5}	—	
Φ_8	$3.194(1) \times 10^{-7}$	3.16×10^{-7}	—	
Φ_9	$4.313(1) \times 10^{-6}$	4.31×10^{-6}	—	
Φ_{10}	$-1.987(2) \times 10^{-6}$	-1.99×10^{-6}	—	
Φ_{11}	$4.32(1) \times 10^{-7}$	—	—	

Table 4.3: Comparison of the scaling function expansion coefficients with values obtained by other methods.

$\Phi(\eta)$	Square lattice	TFFSA [17]	High/Low-T DR [17]	Ext. DR [17]
$\Phi(-5)$	-0.10920919	-0.1092101	-0.1092092	-0.1088626
$\Phi(-4)$	-0.15926438	-0.1592682	-0.1592643	-0.1589421
$\Phi(-3)$	-0.25298908	-0.2529928	-0.2529887	-0.2527417
$\Phi(-2)$	-0.44132564	-0.4413450	-0.4413249	-0.4412136
$\Phi(-1)$	-0.78396650	-0.7839665	-0.7839668	-0.7839576
$\Phi(0)$	-1.19773338	-1.1977330	—	-1.1977320
$\Phi(1)$	-1.38984135	-1.3898410	-1.3898417	-1.3898063
$\Phi(2)$	-1.49305602	-1.4930558	-1.4930566	-1.4929849
$\Phi(3)$	-1.56427320	-1.5642732	-1.5642736	-1.5641727
$\Phi(4)$	-1.61885066	-1.6188506	-1.6188510	-1.6187275
$\Phi(5)$	-1.66324828	-1.6632483	-1.6632485	-1.6631076

Table 4.4: Value of the scaling function with small integer argument.

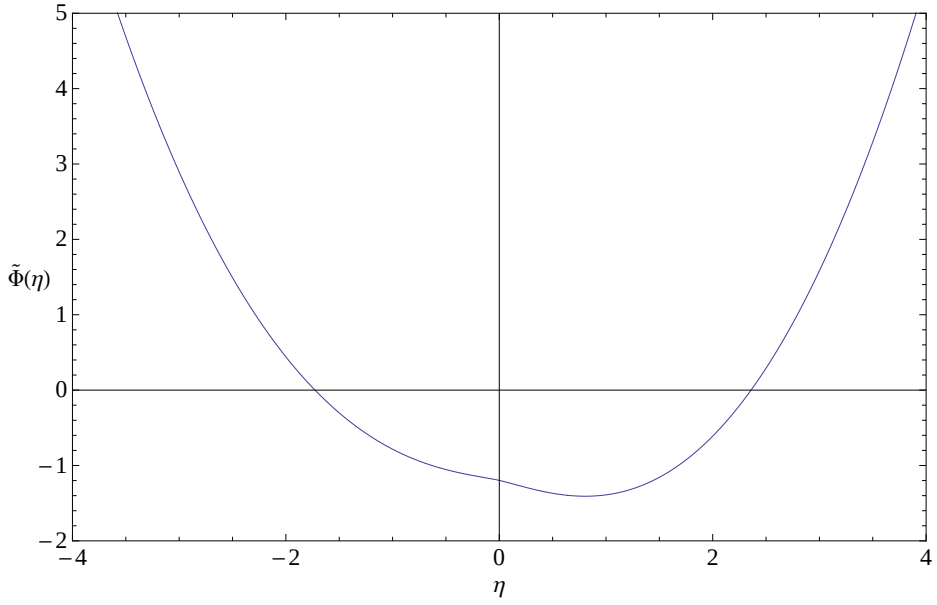

Figure 4.9: Graphic of the regular part $\tilde{\Phi}(\eta)$ of the scaling function.

Chapter 5

Other Models

As stressed several times, the 2D Ising model is one of the most studied models in statistical mechanics, so any additional information about it comes at a high price.

There are a lot of other models, some of them are worth studying, and some information could be obtained or confirmed with much less effort. So as an additional test of the algorithm, some limited study of the three-state Ising and Blume-Capel models has been done. Both models are well studied, but not as thoroughly as the Ising model. The chapter contributes almost no new results, but confirms various results known for these lattice models.

Extension to an arbitrary number of spin states. The CTMRG algorithm can be extended to an arbitrary number N_Z of spin states in a straight forward way. Every spin variable in the A and F matrices now runs over $1 \dots N_Z$. During the construction (2.2) of the extended matrix U, we also let index i run over the desired number of states. Basis transformation and the projection (2.6) are also repeated N_Z times instead of 2.

Obviously, since the extended CTM on each step is N_Z times larger than the original CTM, and there are now N_Z matrices $U(i)$, performance will be noticeably degraded in comparison to the Ising model.

It is not clear how to use the algorithm if N_Z is not a small natural number, as is sometimes considered in more exotic models. For large N_Z that could be used as an approximation for $N_Z \to \infty$ the CTM method is apparently not very feasible.

Tests. Since there are very few references to CTMRG with the number of states more than 2, worth noting is a study of the 5-state Potts model [81]. Due to the fact that the algorithm could fail rather quietly (giving reasonably looking but slightly incorrect results), several tests against exactly solved models were performed.

A natural choice of test model is the (non-chiral) Potts model [82, 20] which has the following Boltzmann weight:

$$w(a,b,c,d) = \exp\beta(\delta_{ab} + \delta_{bd} + \delta_{ac} + \delta_{cd}), \quad a,b,c,d = 1 \dots q \tag{5.1}$$

where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. It has a phase transition with a known temperature, the critical free energy is also known exactly. We calculated the critical free energy for the model with number of states $q = 3, 4, 5$ using double precision algorithms and size $32q$ matrices. Although the Potts model for $q > 4$ and $q < 4$ has rather different properties for the phase transition, the numerical error of the partition function per site κ after several thousand iterations was about 10^{-9}. Qualitatively, for $T < T_c$ all distinct phases with different spontaneous magnetisation phases have been observed. This observation was made mostly for gauging one of the methods for the detection of the critical point, intended for use for the three-state Ising model.

Later it was reported [83] that with some minimal modifications, the same implementation of the algorithm was able to produce correct results for the Kashiwara-Miwa model [7, 84], while the changes needed to do the calculation for the much-discussed chiral Potts model [85] are being investigated.

5.1 Spin-1 Ising model

The model can be seen as a "sister model" of the usual Ising model. It has the same Hamiltonian as the Ising model, but spins σ at the lattice vertices can take three values -1, 0, 1. For the square lattice the partition function looks exactly the same as (1.1)

$$Z = \sum_{\text{states}} \exp\left[\frac{1}{T}\sum_{ij}\sigma_i\sigma_j + H\sum_i \sigma_i\right], \quad \sigma_i = -1, 0, 1, \tag{5.2}$$

with the number of spin states now 3^N, where N is the number of spins. The three-state Ising model is also known as the spin-1 Ising model. The usual Ising model

can be called the spin-$\frac{1}{2}$ Ising model, since for a quantum particle with spin $\frac{1}{2}$ the eigenvalues of $\hat{\sigma}_z$ spin projection operator are $\frac{1}{2}$ and $-\frac{1}{2}$, this corresponds to the usual Ising model by simply changing β and H scales; while for the particle with spin 1 there are three eigenstates of $\hat{\sigma}_z$ with corresponding eigenvalues $-1, 0, 1$.

The model has a critical point with some $T = T_c$ and $H = 0$. The transition temperature is not known exactly, so the first computation could be the determination of T_c. The main source of data for the model is [86], where a substantial study of the model is presented and done mostly using series expansions.

5.1.1 Transition Temperature and the Critical Exponent

The first value that we tried to calculate was the critical temperature. Using only double-precision arithmetic, we "scanned" a range of temperatures, calculating observable values. The range was taken to include the vicinity of the estimate obtained in [86], although it also could be easily obtained independently.

Transition detection. The scanning for the critical point was performed with a simple "bisection" algorithm, the only problem being to choose the value that is maximised exactly at the critical point.

The following values were tried as "markers" of the phase transition:

- The specific heat singularity. We compute $c(T) = \frac{\partial^2 \kappa}{\partial T^2} = \frac{\partial U}{\partial T}$ and then seek the maximum of $c(T)$.
- The magnetisation singularity. We compute $M(T)$ and then seek when it becomes zero.
- Various entries of the CTM, e.g., the second largest eigenvalue, the second largest eigenvalue in a particular $A(i)$, the ratio of the second largest eigenvalues in the different $A(i)$, etc.
- Convergence speed. We count the number of steps that are needed for the speed of convergence to fall below some limit.

But as expected and later verified using the Potts model, all these markers are biased in respect to the real T_c with displacement depending on N and number of iterations N_S, since the iterative solution is only an approximation to the actual solution of (1.45). For the Ising model the bias didn't matter, because no precise calculations were required in the immediate vicinity of the critical point.

To overcome this problem and obtain T_c, one needs to extrapolate to $N \to \infty$ and $N_S \to \infty$. To some extent this is possible, but ultimately due to numerical errors in the source data, the extrapolation is rather unstable, and the precision gain is not enough to improve T_c. The bias effect and possibility of extrapolation wasn't studied in the current work, and its study is rather complicated, although first attempts were made a long time ago [87]. Our best estimate for T is 1.69363 which is to be compared to the estimate $1.69353 - 1.69374$ from [86].

Critical exponents. In [86] the estimate of the critical exponent β and its corresponding subleading ("confluent") exponent was made. Both are defined by an asymptotic expression similar to (3.6):

$$M \sim |t|^{\beta+\Delta|t|+O(t^2)}, \tag{5.3}$$

where $t = T - T_c$. The estimate for β is $0.1248 - 0.12498$. Naturally the conjectured exact β is then $1/8$, the same as for the usual Ising model. A rough estimate for Δ is given as $1.13 < \Delta < 1.4$.

For β, despite the the critical temperature T_c not being known, it seemed possible to make the following non-linear fit:

$$M = |T_c - T|^\beta + \text{dummy terms}, \quad T < T_c. \tag{5.4}$$

Using a general non-linear fit with a varying number of data points and dummy terms one could try to obtain both T_c and β. But all attempts to use such an approach were futile; the fitted function was very unstable to any variation in the input data, mostly because the absolute value function $|\cdot|$ behaves very badly in such types of fitting.

Alternatively, we could substitute T_c obtained using transition detection, and then use a linear fit of $\log M$ to get β:

$$\log M = A + \beta \log |T_c - T| + O([\log |T_c - T|]^2). \tag{5.5}$$

With data point and dummy term variations (as described in Section 4.2.2), plus varying the inexact value of T_c, we estimated β as $0.1235 - 0.1255$, which is in agreement with the above estimate [86] and with the conjecture $\beta = 1/8$.

For the coefficient Δ, the estimate using any of the methods above was not possible even after substituting $1/8$ for β. The only clear bound that we were able to obtain is that $1 < \Delta < 2$.

Summarising, essentially the same results were obtained with a much simpler method and much less effort, although it is clear that the series expansions from [86] provide a solid basis for improvements, while any numerical data would require total recalculation to get better results.

5.2 Blume-Capel Model

The Blume-Capel model was introduced in independent work [88, 89] with two different motivations. It can be seen as an extension of the spin-1 Ising model, and all results for the spin-1 Ising model are applicable to the Blume-Capel model with $\Delta = 0$.

The partition function of the model is

$$Z = \sum_{\text{states}} \exp\left[\beta \sum_{ij} \sigma_i \sigma_j + H \sum_i \sigma_i - \Delta \sum \sigma_i^2\right], \quad \sigma_i = -1, 0, 1. \tag{5.6}$$

Here Δ is the crystal field, H is the external magnetic field and β is the usual inverse temperature. $H = 0$ in all our considerations.

Limits of the model include the spin-1 Ising model when $\Delta = 0$ and the spin-$\frac{1}{2}$ Ising model when $\Delta \to -\infty$.

Not many exact facts are known about the model. Several references worth mentioning are [90, 91], plus some studies using various numerical methods, the main ones being references [92, 93].

Tricritical point calculations. The model has a tricritical point; that is, the point where three phases not related by a simple symmetry transformation (i.e., a transformation defined by the Hamiltonian) can co-exist. The point is situated at the end of the critical line, which goes via critical points of spin-$\frac{1}{2}$ and spin-1 Ising models for $\Delta \to -\infty$ and $\Delta = 0$ respectively, so one might expect that the critical exponent β equals $\frac{1}{8}$ along that line, although β it could change near the tricritical point. The tricritical parameters are not known exactly. Δ for the tricritical point is slightly smaller than 2. Many studies on the tricritical point parameters have been published [92, 93]. The tricritical point estimates given in [93] vary across different methods, so one more independent method to confirm the values of parameters is worth considering.

We performed calculations around the critical point in order to detect critical values of the parameters β and Δ. Detection algorithms were similar to the

ones used for the spin-1 Ising model. In fact we mostly employed a ratio of the CTM entries to detect the phase transition. A search was done using square 20×20 points in (β, Δ) space around the possible tricritical point, with several successive bisections. The tricritical parameters were found to be $\Delta_{3c} = 3.336$, $\beta_{3c} = 1.68634$. The results mostly agree with values given in [92, 93]. More advanced algorithms for finding the extrema could be used, but apparently it is more difficult than searching only by β for case of the spin-1 Ising model, and more numerical calculations are needed.

Conclusion

There are several aspects to mention in concluding this thesis. Apart from the summary of obtained results, we must mention all the research that could be included in this work, but for various reasons was not. Some of these things could actually develop into additional topics of research, possibly giving enough results for many other studies. It is possible that some of the listed suggestions were already developed, but were unnoticed, since the literature on the DMRG-related topics is rather bulky.

Summary of results. The core result of our work is the scaling function of the two-dimensional Ising model. We numerically obtained its series expansions on the three types of regular two dimensional lattice and demonstrated that despite different lattice-dependent contributions, the scaling function is the same for all three lattices.

This clearly demonstrates that two related hypotheses, namely the Scaling Hypothesis and the Universality Hypothesis, hold on the lattice to a very high degree of accuracy, and not just approximately as is sometimes suggested. Several parameters describing non-universal contributions were determined numerically for the first time. The calculations also further confirm the significance of Zamolodchikov's remarkable connection between E_8 CFT and the Ising model in a magnetic field.

We intentionally used as many exact results as were available, thus greatly improving the precision of our numerical results. Not all exact data are absolutely required, but it would be much harder to reach a comparable precision without them.

As an important by-product of this work, a robust implementation of the CTMRG algorithm has been written. After minor modifications it can be adapted for use with many other models.

Possible improvements. There are a number of small things that could be added to the current work without intensive additional research. We give several examples, but the list is by no means exhaustive.

More efficient boundaries of Zones 1, 2, 3 could be calculated and then used for more optimal computations. Convergence detection procedures could also be improved, reducing the amount of calculations required. Using these more efficient computations, for the triangular and honeycomb lattices a better analysis of irrelevant operator contributions could be made.

There are many other models that could be easily fed into the program, but they might lack precise data for their critical parameters and phase diagram. If a more consistent and possibly more automated procedure for computing the scaling function could be found for models like the spin-1 Ising model, (where no data for the exact value of the critical parameters exist, let alone any exact solution), then one could easily produce a "library" of the models with their basic properties, or even a library of scaling functions.

Directions for future research. The first obvious extension is to adapt the algorithm to lattices like the kagome and various other semi-regular lattices, with possible generalisation to lattices with less translation symmetry, including random or not entirely planar lattices.

We mentioned that near the critical point the smoothed singularity is biased with respect to the actual position of the critical point. A better understanding is required as to how critical parameters are changed, how real singularities are smoothed out for finite N, and how to correct for errors introduced by finite CTMs and extrapolate to infinite CTMs as close to the critical point as possible.

Another very important direction for research is extending the algorithm in a way that would make it possible to compute the values of the free energy and its derivatives not just for real-valued H and T, but for complex-valued parameters. This would allow us to directly investigate the structure of the Yang-Lee and Langer singularities which play a crucial role in determining the analytic structure of the scaling function. One cannot just put the complex values in the input data and run the algorithm. In contrast to series expansion where such extension is usually rather easy, algorithms like CTMRG which use a lot of numerical linear algebra machinery, require careful study of all convergence properties and possible elimination of unphysical degrees of freedom.

During the calculation many computation artefacts occurred. One of them drew our particular attention, because it might have some interesting physics be-

hind it. It might also be connected with the scaling function analytical structure research from the previous paragraph. This is the change of the spontaneous magnetisation sign M when the sign of the external field H is changed. We hope it is somehow connected with the decay of the metastable "bubbles" in the non-equilibrium model and related analytical features of the scaling function.

An attempt has been made to employ the algorithm for the Self-Avoiding Walk (SAW) model and several types of Restricted Solid-on-Solid (RSOS) models. The main problem is that the straight forward application of CTMRG to SAW would yield a model with many spin states and big blocks, making practical computations very inefficient. A specialised version of CTMRG should be devised for the SAW-like models. The algorithm would be more restrictive, but could allow us to make more efficient calculations and obtain the scaling function of the SAW model.

At the inception of CTM methods, there were two approaches – fully numerical solutions and series expansions. The numerical solutions then evolved by introducing RG-related concepts, making it one of the most efficient algorithms in this class. It would be very interesting to introduce some of the ideas of CTMRG to the CTM-based series expansion methods.

But by far the most tempting direction for the future research is an efficient 3D version of the algorithm.

Three-dimensional algorithm. In its reduced version a 3D variant of the algorithm was used in [94, 95] for analytic calculations in the Zamolodchikov model. By "reduced" we mean that CTM is $1 \times 1 \times 1$ "tensor" or actually just a number, so it is not necessary to diagonalise it. The reduced form in 2D directly corresponds to the well-known Kramers-Wannier approximation. While in 2D the CTM matrix could be algorithmically diagonalised, no general algorithm exists for the 3D analogue. One could try to solve the CTMRG equations using general non-linear equation solving methods. Alternatively, one might try to find specific symmetries for tensors in the equations, and employ specialised algorithms to diagonalise them.

In [96, 97] a CTMRG version of the reduced algorithm was presented, but no practical improvements were made. At the same time, the original DMRG was extended using so-called tensor networks to represent the state of 1D quantum systems, i.e., their Ψ-function. With some additional conditions in the algorithms like PEPS [55] these tensor network-based methods have been extended to the 2D dimensional quantum models. The ground-state of the 2D quantum models

maps to the equilibrium state of the 3D classical statistical mechanics models, hence the methods could also be potentially used in statistical mechanics. Since the CTMRG is a rather competitive algorithm for 2D systems, it seems rather natural that it should be possible to use it in 3D systems hence obtain interesting and novel results.

Appendix A

Implementation Details

In this Appendix we describe some implementation details of the program that was written and used for the high-precision calculations discussed in this thesis.

While many features of the program were described in the main text, there are still many peculiarities that one should know to successfully employ the program. Most of such details could be deemed to be too technical or non-physical for the main text. Indeed, the style of this Appendix is more programming centred.

Apart from the more or less finished programs documented in this Appendix, there were a lot of programs written for testing purposes. They are not documented but still situated in the source tree. Most of them cannot be used for anything but for the test that they were written for.

It is also worth noting that some features and subprograms described here were not implemented for all variants of the program. The most complete implementation is for the square-lattice Ising model.

A.1 Program Structure

The whole program consists of two main layers: working subprograms written in Fortran [98, 99] for the sake of speed, and various control programs where performance is not important, so they could be written in interpreted language to ease development and use. Such a structure especially facilitates making small frequent changes in the control programs. The latter programs were written mostly in Python [100] without use of any non-standard libraries.

As was stressed several times before, all the main calculations are spread across the parameter space (H-T), and for each point computations could be done in parallel. Each particular way of covering the parameter space for the

model is called a "run".

A.1.1 Variable names

Most of the notable variables are more or less consistently named across all parts of the program. These variables are:

- Hamiltonian parameters and temperature, i.e., all variables defining the weight w: `J0`, `J1`, `J2`, `temperature`, `H0`, `H1` (naming follows the main Baxter paper [36]). In the Fortran part these variables are prefixed by `global_` to avoid collisions with local short-named variables.

- `NB`, `NBMIN`, `NBMAX` – the current size, minimal (starting) and maximal size of the CTM, respectively.

- `NS` – the number of steps (iterations) performed for each `NB`. Note that this could be made dependent on `NB` for performance reasons.

- `NZ` – the number of spin states. For the Ising model this is 2, but for algorithm tests and for the calculations in Chapter 5, this parameter was also set to 3, 4 and 5.

- `A`, `F` – CTM and HRTM. Indexing of A and F was also consistent, with implicit spin indices always going before explicit ones. That is, A is `A(i,a)` where `i=1..NB` and `a=1,2`. Indices for one side of the HRTM are grouped together: F is `F(i,a,j,b)` where `i`,`a` are from one side of the HRTM, and `j`,`b` from the other. If there are several explicit indices on one side, they go from the faster running index to the slowest one (according to the re-enumeration that usually follows construction of such objects, usually the closer an index is to the origin, the slower it runs). If there are explicit indices without implicit counter part (such as in extended CTM H), then they go last.

  The enumeration of spin states runs from the highest spin value to the lowest, to be consistent with the Ising case where 1 is "+"(spin up), and 2 is "–"(spin down).

A.1.2 File System Data Formats

All numerical data (including physical, but not system parameters) is stored in a single global (independent of the run name) directory[1], which is configurable with the `DIR_HDATA` parameter. Names of sub-directories inside that global directory directly correspond to the names of the runs, and the names are then used for issuing all commands doing computations with data from the run. All data (both input and output) for the run is contained within its directory, allowing for quick renaming and moving the runs around. All other mentions of files and directories (except for global configurations and building the programs) imply that the files are in the run directory.

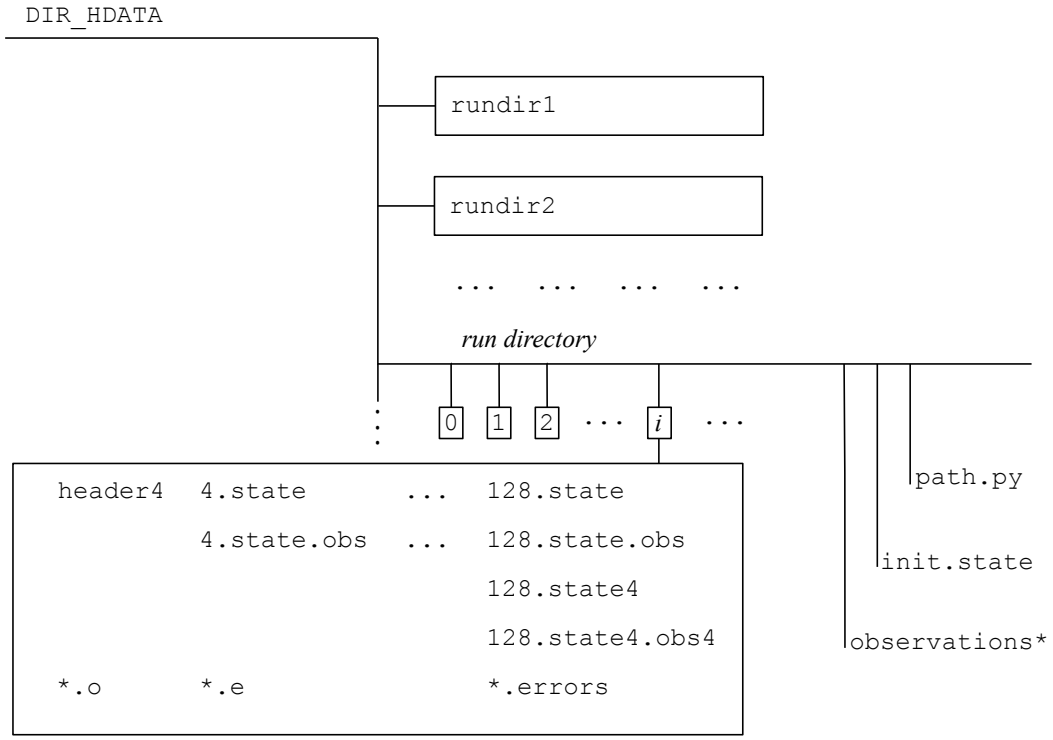

Figure A.1: File system data formats.

Run directory structure. Initially only the `path.py` file is in the run directory. Optionally `init.state` could also be there as an initial state (in a "state-file" format). After performing the `prepare` command there are sub-directories corresponding to the points in the parameter space, the names of these directories are integer numbers counting from 0. All files with numerical data, values of A

[1]Normally it was in the home directory and called `data2` (signifying the second version of data format).

and F are stored inside the sub-directories. To collect observable values the user issues the `collect` (or `collect4`) command, and observable values are gathered in the file called `observables.*`, where `*` indicates various suffixes, such as `4` for QUAD precision `observables`. The format of `observables.*` is a CSV (Comma Separated Values [101]), the first line indicates the names of the columns (note that the order of the columns is undefined).

Files formats. Inside every "point" subdirectory there are two main types of file. One of the types is formatted data, composed of arrays and namelists [99], the other type is various text files with diagnostic logs. The usual files are:

- `header`, `header4` – These files contain two namelists (`HAMILTONIAN` and `TEMPERATURE`) that set all Hamiltonian parameters and temperatures.
- `*.state`, `*.state4` – or "state-files". These consist of `NBNL` namelists with a single variable `NB` (line 1) and arrays `A` and `F` written in the formatted Fortran notation (lines 2 and 3).
- `*.state.obs`, `*.state4.obs4` – contain `OBSERVABLES` namelists.
- `*.state.errors`, `*.state.obs.errors` – log files, `stderr` output of `grow` and `observables` stages.
- `*.o`, `*.e` – PBS log files [102]. For example, they contain warnings issued by control scripts themselves. `*` in these names are names of the PBS jobs (set by `runjob.py` script); two files are for `stdout` and `stderr` respectively.

For all these files (except for the last one) `*` in the name is `NB`. Usually, only states with `NB` equal to `NBMIN` and `NBMAX` are saved, but if the configuration parameter `SAVE_STATES` is set to `True`, states are saved for all values of `NB`.

A.1.3 Subprograms

Subprograms are naturally divided by the layer of the program they belong to. Computational subprograms are written in Fortran, and do not have any command-line parameters. They are compiled, not interpreted, and all of them don't read or write any files; all input-output is done via `stdin` and `stdout` (non-critical messages are printed to `stderr` and redirected to `*.errors` files mentioned above). In the control subprograms standard input and output are

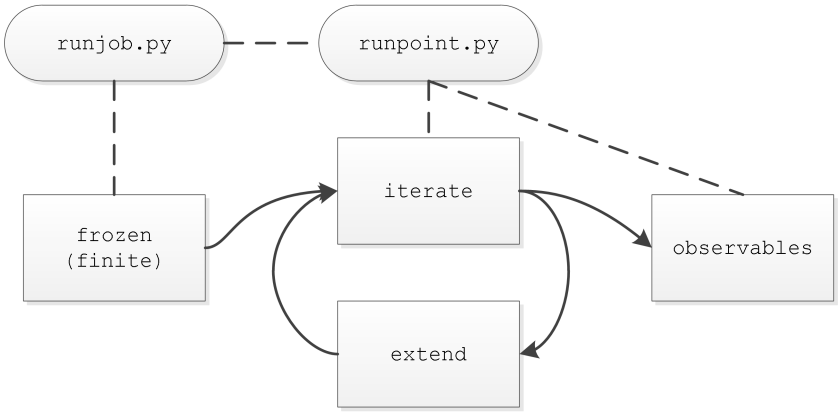

Figure A.2: Program structure.

not used extensively (the latter is used for information messages though), the parameters are supplied via command-line options and configuration files.

Computational subprograms are:

- `iterate` – this is by far the most important subprogram. It reads sequentially: `HAMILTONIAN` and `TEMPERATURE` namelist to construct the Boltzmann weight; `NSNL` namelist with the number of iterations, and then the state (represented as a "state-file"). After reading, it performs CTMRG iteration the desired number of times and outputs the "state-file" to the standard output.

- `extend` – the subprogram reads the "state-file" data, and then outputs the same data with `NB` increased by one. New rows and columns of matrices are filled with zeros.

- `observables` – reads `HAMILTONIAN` and `TEMPERATURE`, then "state-file", and computes observable values, outputting them as a namelist. In the current version of the program the choice of values to compute is hard-coded, the user can keep several versions of the `observables` subprogram, switching to the desired version if different sets of observable values are to be computed.

- `frozen` – this subprogram takes only the `NB` namelist as an input, and outputs the constructed state that roughly corresponds to the $T = 0$ temperature. For the Ising model the trick works quite well, but for other models convergence to the solution depends on a good enough initial state, so `exacts` was used.

- `exacts` – is used when `frozen` wasn't enough for good convergence. First, from the `stdin` the subprogram reads the `LNL` namelist with the value of `L`, and then the standard namelists `NBNL`, `HAMILTONIAN` and `TEMPERATURE`. Then it constructs an exact (for the finite lattice) CTM and corresponding HRTMs with the size `L`, diagonalises the CTM (converting HRTMs to the proper basis) and takes its `NB` highest eigenvalues making the state ready to be written to `stdout` in "state-file" format.

  This subprogram was mostly used for the Blume-Capel model, where convergence with the `frozen`-generated state wasn't satisfactory in some regions of the parameter space. In some similar situations a hand-crafted `init.state` file was also used.

- `2to4` – a small utility subprogram that converts the DOUBLE "state-file" to the QUAD "state-file". Please note that all additional digits that appear in the output don't have any physical sense and in fact depend on the machine representation of the floating-point numbers.

There are only two main control scripts, but unlike the computational layer subprograms they each perform a bunch of functions. These functions are considered in a separate section A.2 from a more user-centred point of view, including a list of subcommands and runs-related configuration.

- `runjob.py` *run_name subcommand subcommand_parameters* – this script governs "global" tasks, i.e., tasks that are performed with all points of the run. Depending on the global configuration, the script could run some subcommands on the cluster in parallel (and thus asynchronously) using the batch system.

- `runpoint.py` *run_name subcommand point_number* – executes subcommands local to the point and has a fixed number of arguments. In most cases the script is run in multiple instances (enumerated by *point_number*) on cluster nodes. All sub-tasks of the script are simply its sub-processes, they run on the same node in the cluster environment, so the script doesn't return until they have completed.

A.1.4 Global Parameters

The program successfully runs on different computers and clusters, with the configuration for the particular system being relatively easy. There are many "global"

parameters, such as paths to files and parameters that do not vary between different runs. Only a few such parameters need to be set up, others have usable default values. All configuration files are essentially Python scripts that could contain arbitrary Python code that only needs to set the values of several local variables with predefined names; this feature is actually used to access the environment variables, such as `$HOME` and `$PBS_JOBFS`.

There are two files that are used for setting global parameters. The first one with all default values is named `siteconfig.py` and is located in the same directory as `runjob.py` and `runpoint.py` scripts. In the end of the standard `siteconfig.py` an attempt is made to execute a file located in the home directory named `locals.py`. The `locals.py` file usually contains parameters that are changed more or less frequently. If no `locals.py` was found then every control script prints a warning at the beginning of its execution.

The following parameters are worth noting:

- Parameters dictating where various things are located:
  - `DIR_HOME`, `DIR_HDATA`, `DIR_PROGS` – the home directory, main data directory and a directory with executables. Only the second variable is used directly in the program, others are defined for convenient construction of other path variables.
  - `CMD_ITERATE`, `CMD_EXTEND`, `CMD_OBSERVABLES`, `CMD_FROZEN`, `CMD_FINITE`, `CMD_2TO4` – a set of parameters telling where executable files of the program are located. These could be used for easy switching between various versions of compiled executables.
  - `CMD_QSUB`, `CMD_RUNPOINT` – give the locations of two "special" executables: PBS's `qsub` command [102] and `runpoint.py` script.
- `tmpl` – PBS script template [102]. It should contain three `%` as placeholders for the job's name and paths to log files for `stdout` and `stderr`. Other job parameters, such as `walltime`, `vmem`, `ncpus` [102, 103] are currently hard-coded.
- Various internal parameters:
  - `SAVE_STATES` – if `True`, states with all `NB` are saved.
  - `USE_QUAD` – describes whether to use QUAD precision by default in `grow` and `observables` commands.

– `SHORTHAND_FACTOR` – an optimisation hack: if `NB` is larger than `NBMAX/2`, then the number of steps `NS` is divided by this factor.

In the supplied `siteconfig.py` there are several other variables, but they are currently unused and don't affect any aspect of the computations.

A.1.5 QUAD Precision Notes

Almost from the very beginning of the practical calculations it was clear that the usual hardware floating-point numbers are not precise enough to obtain the scaling function from the lattice data with sufficient accuracy. So the precision of all data needed to be improved. There are libraries that implement arbitrary precision floating-point arithmetic, and many computer-algebra systems, including Wolfram Mathematica [75], that support arbitrary-precision numbers natively. Fortran also formally supports arbitrary precision (via `KIND` specification [99]), but by the standard the compilers are not required to support intrinsic function for any unusual kind of floating point numbers.

But of course the core problem in using arbitrarily large precision is degraded performance. In comparison to the hardware arithmetic, emulated arbitrary precision calculations are too slow to do any serious linear algebra with them. The compromise is to use QUAD precision numbers [104], which are defined by the IEEE standard and are partially supported by modern CPUs (Intel Itanium 2 CPUs have full support [105]). They are only several times slower than the usual double precision numbers. Each QUAD number occupies 16 bytes and has 33-34 significant digits in the mantissa. The Intel Fortran Compiler fully supports QUAD numbers with just one command-line switch [98] that allows building QUAD-enabled programs without any changes to the computational part of the source code. Changes are still required for the I/O part, but in the current version of program correct `FORMAT` specifications for I/O subroutines are selected automatically.

Until the latest release, GNU Fortran didn't support QUAD precision. The situation has changed in the very recent version – GCC 4.6 [106], but no testing of GNU Fortran with such precision was made.

Since the standard Intel Math Kernel Library, as well as many other precompiled libraries, don't support QUAD numbers, custom versions of required linear algebra routines were compiled. For the eigenvalue decomposition EISPACK [107] was used, mainly because of its simplicity and ease of "extracting" the required

subroutines from the library. For SVD the subset of Reference LAPACK [51] was used, including several dozen subroutines which `DGESDD` depends on.

Some linear algebra algorithms work correctly only with specific "quirks" of floating point arithmetic. We tried to use the most generous algorithms (such as Divide-and-Conquer for the SVD [44]) to avoid any potential trouble with emulated QUAD arithmetic, which could have more deviation from the standard than the hardware implementations, mostly for the sake of performance. While the correctness of the algorithms wasn't tested in full, the ultimate test – equality of the numerical solution with the exact solution when the latter exists – was passed perfectly.

A.1.6 Source Code Organisation and Program Building

In the previous sections subprograms and scripts were described without consideration of the place where they reside or how one compiles and links ("builds") them. Since the program was not intended for distribution, some parameters are switched by changing the Fortran code and then recompiling the subprograms.

Source Code Organisation. The following files and directories reside in the main directory:

- `adapters/` – EISPACK files needed for eigenproblem solving and the adapter file to match interface of LAPACK `DSYEVD` subroutine.
- `bs/` - "build scripts", contains all scripts using for building the program.
- `lapack/` – the reference LAPACK subset, all files needed for `DGESDD` function.
- `old/` - old and testing codes, not used.
- `runner1/` – the first version of control scripts, not used, but was kept for reference while writing the second version.
- `runner2/` – the current version of the control scripts. Apart from `runpoint.py`, `runjob.py` and `siteconfig.py` scripts, there are common files that the scripts depends on, plus many old/specialised versions of the scripts.
- `src/` – the main directory for the Fortran source codes. Most files here belong to two categories: named `bxt_*.f90` and `*_{sym,asym,hc}.f90`.

- `compopts.*` (some of them are also in `bs/` directory) – shell scripts with different compilation and linking options. The active file is chosen by a symlink with name `compopts.local` in the home directory.
- `suffix.*` – shell scripts with current "suffix" – {`sym,asym,hc`}. This contains definition of one variable `SUFFIX`. Together with specific naming and structure of the Fortran source code files, this rather primitive system allows for compile-time switching of the algorithms used for particular calculation (e.g., switching between symmetrical, asymmetrical and honeycomb lattices). Unfortunately, the system has many deficiencies, and source code editing and recompiling is sometimes required to change the active algorithm (or having precompiled sets of files for each algorithm). The active file is chosen by a symlink named `suffix.local` in the home directory.
- `*.o`, `*.mod` – object and module files are currently stored in the main directory, so during or after the build the main directory contains many such temporary files.

There are many others files, that are either auxiliary or just temporary.

Building the Program. Due to the very nature of the original program purpose, the process of building the program wasn't a target for optimisation or significant automation. Instead of using some build system, such as GNU Make or SCons (both are quite complicated but powerful and flexible), all compilation and linking is done via several shell scripts [108]. The set of compilation parameters are fast-switchable and tuneable [98] via special configuration files `compopts.*`. Several configuration procedures such as switching between the lattice types are done via changing the source code and then recompiling the program. Also note that most of these scripts assume that temporary files are in the current directory, so they should be run from the main program directory.

The main scripts used to build the program are:

- `compile` *file*, `compile4` *file* – compile *file*. Note that `file` is a real filename with path (usually `src/`) and extension (`.f90` or `.f`).
- `link` *file*, `links` *file*, `link4` *file* - the scripts to link executables. Filename *file* doesn't include the path or extension or suffix. For example, to link the `iterate` subprogram one issues a `link iterate` command. Note that all object files that are required need to be already compiled, i.e., the scripts don't perform any compilation.

- `build`, `builds`, `build4` – these scripts call `compile` and `link` scripts to build double precision, statically linked double precision (to alleviate some problems with Intel MKL on certain clusters) and QUAD precision versions of the program respectively. Before the build they delete all object and module files from the main directory.

- `2to4compile` *file* – a special script for building the `2to4` subprogram. *file* is the path to `2to4.f90` source file (usually just `src/2to4.f90`). The reason for the separate building script is that `2to4` is the only Fortran executable that uses both double and QUAD precision numbers and hence it requires special treatment. The same argument accounts for `2to4` not having any dependencies; it is built from a self-contained source file `2to4.f90`.

- `build_lapack4` – compiles a SVD-related subset of LAPACK from `lapack/` to use with the QUAD precision version of the program.

To fully utilise the QUAD precision option the program essentially needed to be compiled twice, in double and QUAD precision versions, plus a "bridge" subprogram `2to4`. Ideally, this is done just by three commands `build`, `build4` and `2to4compile`. All `link` scripts place executables into the `BINDIR` directory, which is defined among other options in `compopts.*` files. Usually it is the same directory as `DIR_PROGS` in `siteconfig.py` options.

Another configuration option to note is `suffix`. It governs which files are chosen by `link` scripts; e.g., `iterate` executable could be built from `iterate_sym.f90`, `iterate_asym.f90` or `iterate_hc.f90`. Also note that these source files contain nothing but a single call to the subroutine in `bxt_iterations.f90` file. Many functions and subroutines also have the same style of suffixing.

Compilation Option. Files with compilation options are just `bash` scripts defining several variables. The main variables are:

- `FC` – the command to run the Fortran compiler, without explicit path (`PATH` environment variable is used). For example, `gfortran` or `ifort`.

- `OPTS` - provides options to the compiler, e.g., optimisation or debug options.

- `BINDIR` – the destination directory for the executables.

- `LIBS` – contains options for linking, usually a set of `-l`*name* options to link libraries. Note that Intel MKL options are quite complicated, and there is a special tool to help decide which options to use [109].

- `LIBDIRS` – contains options indicating where the libraries reside, in `-L` */path/to/dir* form

Several sets of compilation options were used, all of them (except the last one) use the Intel Fortran compiler [98]

- `compopts.dbg.64`, `compopts.dbg.32` – these are sets of options mainly for debug, without any performance-improving flags. `32` and `64` in the names indicate 32- or 64- bit CPU architecture; in most other cases only 64-bit architecture is used.

- `compopts.moderate` – moderately optimising options, the most used configuration.

- `copopts.opt` – optimising options. Note that it contains parameters that might not be safe in the sense that the compiled program may produce wrong results.

- `compopts.gfortran` – to use the GNU Fortran compiler and self-compiled reference LAPACK library where Intel Fortran and MKL are not available. Options mostly correspond to `compopts.moderate`; they were tested on Mac OS X 10.6 with GCC 4.5.

In future versions of the program all this machinery will be replaced by Makefiles (perhaps also with CMake help).

A.2 User Manual

In this section we discuss usage of the (compiled and configured) program. All run-specific configurations is defined in a single file `path.py`, so performing a run usually means configuring `path.py` and then using `runjob.py` to calculate it.

`path.py` **format.** The file is a python script that defines just two variables: `numpoints` and `points`. `points` is the main variable, and is an array of Python `dict` objects that describe points in parameter space. `numpoints` is nothing but the length of that array. Note that in the conventional `path.py` there is code that sets common values for all dictionaries and automates the distribution of points across the parameter space.

The dictionary for each `points` element contains the following keys:

- `NBMIN`, `NBMAX` – defines minimal and maximal `NB` values for the point. In the current implementation it is required that the `NBMIN` be the same for all points, while uneven `NBMAX` could affect the `collect` command only.

- `START` – says which initial state to use in the very beginning of iterations for the point, when `NB` equals `NBMIN`. Possible values are: some number to use as an index of the specific entry in the `points` array (note that the index should be less than the current point index, since in the beginning points are always computed sequentially); string `previous` – use the previous point in `points` array; string `frozen` – use `frozen` subprogram to generate an initial state; string `exact` – use `exact` subprogram; string `init` – load `init.state` from the run directory.

- `J1`, `JP`, `JPP1`, `H0`, `TEMPERATURE` – Hamiltonian parameters. In practice `START` and Hamiltonian parameters are the only parameters that vary across the `points` array.

- `NS` – the number of iteration steps for each `NB` value. Typically it is around 200-400.

- `NBSTEP` – the increment of `NB` after performing `NS` steps with the current `NB`. Originally its value was 1, i.e. iterations were done for all $\texttt{NB} \leq \texttt{NBMAX}$. Later it was found that `NBSTEP` values less than or equal 4 are also usable.

Note that there is the following problem in the current version. If $\texttt{NBMAX} - \texttt{NBMIN} \neq 0 \mod \texttt{NBSTEP}$ then the actual state will have size large than `NBMAX`, but the filename for the "state-file" will still be {*value of* `NBMAX`}`.state`.

`runjob.py prepare` **command.** This command reads the `path.py` file and sequentially (without running `runpoint.py`) calculates each point with $\texttt{NB} = \texttt{NBMIN}$ using `NS` steps. Calculation progress is printed to the standard output.

After the `prepare` command has completed, it is possible to run computations in parallel. All parallelised computations are run via `runjob.py` *run-name* `points` *subcommand pointnums*, that runs appropriate number of `runpoint.py` instances via the PBS `qsub` command. *pointnums* could be specified as a set of ranges *a-b,c-d,e-f,...*, or the shortcut `all` can be used.

`runpoint.py` **commands.** These are:

- `grow` – the main iteration command; performs an extend-iterate loop until `NBMAX` is reached. It prints progress repots to `stderr`.
- `observables` – calculates observable values for the point. Observables are calculated for each `*.state` file found in the point directory.
- `grobs` – combines `grow` and `observables` into the single command.

Each of these commands has a variant with `4` appended (i.e., `grow4`, `grobs4`, `observables4`) to do calculations with QUAD precision even if `USE_QUAD` is not set to `True`.

`runjob.py collect` command. After computation, observable values are scattered across several thousand files in the points directories. The command `runjob.py collect` *points* gathers these values into the one file in the run's root directory. *points* argument is optional. If omitted then all points are used. Values are collected from {*value of* `NBMAX`}`.state.obs` files. There is also a QUAD variant `collect4`.

A word of warning: existing output files for all of the commands are unconditionally overwritten.

A.3 Known Issues and Limitations

It is doubtful that altering the current program could possibly influence the physical results described in this work.[2] Nevertheless, there are a lot of known deficiencies in the program, many of which were already mentioned. Here we quickly list some bugs and misfeatures that are known, but didn't get enough attention to get fully fixed. Several bugs could be fixed rather easily, but others might require some redesign of the program.

Algorithm and lattice switching. There is no way of changing the lattice type or calculation algorithm without program recompilation. `suffix`-system could be extended to the executables, but this might only increase the mess with switching. In practice the problem was not very important, since only one type of lattice at a time was calculated on a massive scale, and all comparisons of different algorithms and lattices were made using custom programs.

[2] Although some versions of `finite` subprogram have bugs, this doesn't impede the main iteration scheme and their results.

Observables choice and collection. In the current version of the program, a choice of which observable values to compute is made at compile time. One can compile several version of the `observables` executable and then use the appropriate version (e.g., by changing the parameter `CMD_OBSERBABLES` or symlinking). Originally that was a result of "optimisation"; many parts of the free energy and internal energy calculations are shared, so it is better to do them in one subprogram. Later it became apparent that for the different points it is better to have different sets of observable values to be computed.

Error handling and indication. Most errors are handled in a very radical way of program termination. For the parallelised part it means that if some points have failed for any reason, they must be found and rerun. There should be more ways to detect failed points and reasons for the failure, other than manually checking logs.

Better PBS queue management. Currently the only way of managing the PBS queue is adding ranges of `runpoint.py` tasks to it. The user should monitor the queue themselves, and run `collect` command only when all jobs have been completed. The problem is aggravated by the per-user queue limit at the NCI facilities [103], so additional scripts were used to submit `runpoint.py` jobs gradually.

PBS template parameterisation. The only "automated" arguments given to the `qsub` command are related to the name of the job. Memory and walltime parameters need to be command- and run- specific, and ideally their values should be estimated fully automatically.

More parameters must be made run-specific. As many run parameters as possible should be made run-specific, i.e., it should be possible to override any parameter from `siteconfig.py` in the `path.py` and/or by command-line options.

Number of Iterations There are plenty of ways to estimate how many iterations are required for the algorithm to converge; many of the ways were actually implemented and tried. Measures of convergence include differences of CTM eigenvalues or observable values between iterations, but none of these were found to be reliable. Despite this, some convergence estimation methods were used in the attempt to get the critical temperature of the spin-1 Ising model, although

without gaining any additional digits. But for the Ising model the `NS` value was estimated manually by trial-and-error methods, and then hardcoded into `path.py`.

Better integration with Mathematica. The CSV format is a universal and well-understood data format that has many advantages. But for more automated computations with more possibilities to analyse not only the final results, but also the iteration process (and thus have more chances to improve it), a better "bridge" between the program and Mathematica (or whatever program is used for further analysis) is needed. This includes replacement of the CSV with something like HDF5 [110]. Possible improvements also include the ability to run the program directly from Mathematica.

Bibliography

[1] W. Lenz. Beiträge zum verständnis der magnetischen eigenschaften in festen körpern. *Physikalische Zeitschrift* **21**, 613–615 (1920).

[2] E. Ising. Beitrag zur Theorie des Ferromagnetismus. *Z. Phys.* **31**, 253–258 (1925).

[3] L. D. Landau and E. M. Lifshitz, *Statistical Physics*. Pergamon, Oxford, 1968.

[4] L. Onsager. Crystal statistics. I. A two-dimensional model with an order-disorder transition. *Phys. Rev.* **65**, 117–149 (1944).

[5] E. H. Lieb. Residual entropy of square ice. *Phys. Rev.* **162**, 162–172 (1967).

[6] R. J. Baxter. Partition function of the eight-vertex lattice model. *Ann. Physics* **70**, 193–228 (1972).

[7] M. Kashiwara and T. Miwa. A class of elliptic solutions to the star-triangle relation. *Nuclear Physics B* **275**, 121–134 (1986).

[8] M. Jimbo, T. Miwa, and M. Okado. Solvable lattice models with broken Z_N symmetry and Heckes indefinite modular-forms. *Nucl. Phys. B* **275**, 517–545 (1986).

[9] M. Gell-Mann and F. E. Low. Quantum electrodynamics at small distances. *Phys. Rev.* **95**, 1300–1312 (1954).

[10] M. E. Fisher. The theory of equilibrium critical phenomena. *Reports on Progress in Physics* **30**, 615 (1967).

[11] A. A. Belavin, A. M. Polyakov, and A. B. Zamolodchikov. Infinite conformal symmetry in two-dimensional quantum field theory. *Nucl. Phys. B* **241**, 333–380 (1984).

[12] L. A. Takhtajan and L. D. Faddeev. The quantum method for the inverse problem and the XYZ Heisenberg model. *Uspekhi Mat. Nauk* **34**, 13–63 (1979).

[13] L. D. Faddeev, E. K. Sklyanin, and L. A. Takhtajan. The quantum inverse problem method. 1. *Theor. Math. Phys.* **40**, 688–706 (1980).

[14] S. R. White. Density-matrix formulation for quantum renormalization-groups. *Phys. Rev. Lett.* **69**, 2863–2866 (1992).

[15] T. Nishino and K. Okunishi. Corner transfer matrix renormalization group method. *J. Phys. Soc. Japan* **65**, 891–894 (1996).

[16] T. Nishino and K. Okunishi. Corner transfer matrix algorithm for classical renormalization group. *J. Phys. Soc. Japan* **66**, 3040–3047 (1997).

[17] P. Fonseca and A. Zamolodchikov. Ising field theory in a magnetic field: analytic properties of the free energy. *J. Stat. Phys.* **110**, 527–590 (2003).

[18] A. B. Zamolodchikov. Integrals of motion and S-matrix of the (scaled) $T = T_c$ Ising model with magnetic field. *Int. J. Mod. Phys. A* **4**, 4235–4248 (1989).

[19] M. Caselle, M. Hasenbusch, A. Pelissetto, and E. Vicari. Irrelevant operators in the two-dimensional Ising model. *J. Phys. A* **35**, 4861–4888 (2002).

[20] R. J. Baxter, *Exactly Solved Models in Statistical Mechanics*. Academic Press Inc., 1982.

[21] H. A. Kramers and G. H. Wannier. Statistics of the two-dimensional ferromagnet. Part I. *Phys. Rev.* **60**, 252–262 (1941).

[22] B. Kaufman. Crystal statistics. II. partition function evaluated by spinor analysis. *Phys. Rev.* **76**, 1232–1243 (1949).

[23] R. M. F. Houtappel. Order-disorder in hexagonal lattices. *Physica* **16**, 425–455 (1950).

[24] N. V. Vdovichenko. A calculation of the partition function for a plane dipole lattice. *Zh. Eksp. Teor. Fys.* **47**, 715–719 (1964).

[25] R. J. Baxter and I. G. Enting. 399th solution of Ising-model. *J. Phys. A* **11**, 2463–2473 (1978).

[26] R. J. Baxter. On solving the Ising-model functional relation without elliptic functions. *Physica A* **177**, 101–108 (1991).

[27] R. J. Baxter. Corner transfer matrices of eight-vertex model. I. low-temperature expansions and conjectured properties. *J. Stat. Phys.* **15**, 485–503 (1976).

[28] R. J. Baxter. Corner transfer matrices of eight-vertex model: II. Ising-model case. *J. Stat. Phys.* **17**, 1–14 (1977).

[29] L. Onsager. Discussion remark (spontaneous magnetisation of the two-dimensional Ising model). *Nuovo Cimento (Suppl.)* **6**, 261 (1949).

[30] C. N. Yang. The spontaneous magnetization of a two-dimensional Ising model. *Phys. Rev.* **85**, 808 (1952).

[31] T. T. Wu, B. M. McCoy, C. A. Tracy, and E. Barouch. Spin-spin correlation functions for the two-dimensional Ising model: exact theory in the scaling region. *Phys. Rev. B* **13**, 316–374 (1976).

[32] W. P. Orrick, B. Nickel, A. J. Guttmann, and J. H. H. Perk. The susceptibility of the square lattice Ising model: new developments. *J. Stat. Phys.* **102**, 795–841 (2001).

[33] Y. Chan, A. J. Guttmann, B. G. Nickel, and J. H. H. Perk. The Ising susceptibility scaling function. arXiv:1012.5272.

[34] R. J. Baxter. Dimers on a rectangular lattice. *J. Math. Phys.* **9**, 650–654 (1968).

[35] R. J. Baxter. Corner transfer matrices in statistical mechanics. *J. Phys. A* **40**, 12577–12588 (2007).

[36] R. J. Baxter. Variational approximations for square lattice models in statistical mechanics. *J. Stat. Phys.* **19**, 461–478 (1978).

[37] L. P. Kadanoff. Scaling laws for Ising models near T_c. *Physics* **2**, 263 (1966).

[38] A. M. Polyakov. Microscopic description of critical phenomena. *JETP* **28**, 533 (1969).

[39] K. G. Wilson. Renormalization group methods. *Adv. Math.* **16**, 170–186 (1975).

[40] K. G. Wilson. The renormalization group: Critical phenomena and the Kondo problem. *Rev. Mod. Phys.* **47**, 773–840 (1975).

[41] S. R. White and R. M. Noack. Real-space quantum renormalization-groups. *Phys. Rev. Lett.* **68**, 3487–3490 (1992).

[42] S. R. White. Density-matrix algorithms for quantum renormalization-groups. *Phys. Rev. B* **48**, 10345–10356 (1993).

[43] K. A. Hallberg. New trends in density matrix renormalization. *Adv. Phys.* **55**, 477–526 (2006).

[44] J. W. Demmel, *Applied numerical linear algebra*. SIAM, Philadelphia, 1997.

[45] H. Au-Yang and J. H. H. Perk. Ising correlations at the critical-temperature. *Phys. Lett. A* **104**, 131–134 (1984).

[46] S. K. Tsang. Corner transfer matrices of the triangular Ising model. *J. Stat. Phys.* **17**, 137–152 (1977).

[47] R. J. Baxter. Corner transfer-matrices of the chiral Potts-model: II. The triangular lattice. *J. Stat. Phys.* **70**, 535–582 (1993).

[48] R. J. Baxter. Planar lattice gases with nearest-neighbor exclusion. *Ann. Comb.* **3**, 191–203 (1999).

[49] A. Malakis. Phase diagrams of the triangular Ising antiferromagnet-variational approximations. *J. Phys. A* **14**, 2767 (1981).

[50] A. Malakis. Two- and three-spin triangular Ising model: Variational approximations. *J. Stat. Phys.* **27**, 1–17 (1982).

[51] LAPACK numerical linear algebra library. `http://www.netlib.org/lapack/`.

[52] M. Fannes, B. Nachtergaele, and R. Werner. Finitely correlated states on quantum spin chains. *Comm. Math. Phys.* **144**, 443–490 (1992).

[53] S. Östlund and S. Rommer. Thermodynamic limit of density matrix renormalization. *Phys. Rev. Lett.* **75**, 3537–3540 (1995).

[54] G. Vidal. Class of quantum many-body states that can be efficiently simulated. *Phys. Rev. Lett.* **101**, 110501 (2008).

[55] F. Verstraete and J. I. Cirac, Renormalization algorithms for quantum-many body systems in two and higher dimensions. 2004. arXiv:cond-mat/0407066.

[56] A. M. Polyakov. Conformal symmetry of critical fluctuations. *JETP Lett.* **12**, 381 (1970).

[57] V. V. Bazhanov, B. Nienhuis, and S. O. Warnaar. Lattice Ising model in a field: E_8 scattering theory. *Phys. Lett. B* **322**, 198–206 (1994).

[58] R. Coldea, D. A. Tennant, E. M. Wheeler, E. Wawrzynska, D. Prabhakaran, M. Telling, K. Habicht, P. Smeibidl, and K. Kiefer. Quantum criticality in an Ising chain: Experimental evidence for emergent E_8 symmetry. *Science* **327**, 177 (2010).

[59] B. McCoy and T. T. Wu, *The Two-Dimensional Ising Model*. Harvard University Press, 1973.

[60] V. A. Fateev. The exact relations between the coupling constants and the masses of particles for the integrable perturbed conformal field theories. *Phys. Lett. B* **324**, 45–51 (1994).

[61] V. V. Mangazeev, M. T. Batchelor, V. V. Bazhanov, and M. Yu. Dudalev. Variational approach to the scaling function of the 2D Ising mode in a magnetic field. *J. Phys. A* **42**, 042005 (2009).

[62] M. Caselle, M. Hasenbusch, A. Pelissetto, and E. Vicari. High-precision estimate of g_4 in the 2D Ising model. *J. Phys. A* **33**, 8171–8180 (2000).

[63] M. Caselle, M. Hasenbusch, A. Pelissetto, and E. Vicari. The critical equation of state of the two-dimensional Ising model. *J. Phys. A* **34**, 2923–2948 (2001).

[64] A. Aharony and M. E. Fisher. Universality in analytic corrections to scaling for planar Ising models. *Phys. Rev. Lett.* **45**, 679–682 (1980).

[65] A. Aharony and M. E. Fisher. Nonlinear scaling fields and corrections to scaling near criticality. *Phys. Rev. B* **27**, 4394–4400 (1983).

[66] B Nickel. On the singularity structure of the 2D Ising model susceptibility. *J. Phys. A* **32**, 3889–3906 (1999).

[67] M. Caselle and M. Hasenbusch. Critical amplitudes and mass spectrum of the 2d Ising model in a magnetic field. *Nucl. Phys. B* **579**, 667–703 (2000).

[68] K. Okunishi, Y. Hieida, and Y. Akutsu. Universal asymptotic eigenvalue distribution of density matrices and corner transfer matrices in the thermodynamic limit. *Phys. Rev. E* **59**, R6227–R6230 (1999).

[69] Raw numerical data for the 2d ising model on the regular lattices. `http://tpsrv.anu.edu.au/Members/dudalev/thesis-data`.

[70] Numerical solution of partial differential equations. `http://reference.wolfram.com/mathematica/tutorial/NDSolvePDE.html`.

[71] B. Fornberg. Calculation of weights in finite difference formulas. *SIAM Review* **40**, 685–691 (1998).

[72] Online encyclopedia of integer sequences, A000041. `http://oeis.org/A000041`.

[73] Online encyclopedia of integer sequences, A000009. `http://oeis.org/A000009`.

[74] V. V. Mangazeev, M. Yu. Dudalev, V. V. Bazhanov, and M. T. Batchelor. Scaling and universality in the two-dimensional Ising model with a magnetic field. *Phys. Rev. E* **81**, 5 (2010).

[75] Wolfram Mathematica documentation. `http://reference.wolfram.com/`.

[76] C. A. Tracy and B. M. McCoy. Neutron scattering and the correlation functions of the Ising model near T_c. *Phys. Rev. Lett.* **31**, 1500–1504 (1973).

[77] B. M. McCoy and T. T. Wu. Two-dimensional Ising model near T_c: Approximation for small magnetic field. *Phys. Rev. B* **18**, 4886–4901 (1978).

[78] S. B. Rutkevich. Decay of the metastable phase in $d = 1$ and $d = 2$ Ising models. *Phys. Rev. B* **60**, 14525–14528 (1999).

[79] S. Zinn, S. Lai, and M. E. Fisher. Renormalized coupling constants and related amplitude ratios for Ising systems. *Phys. Rev. E* **54**, 1176–1182 (1996).

[80] V. Fateev, S. Lukyanov, A. Zamolodchikov, and Al. Zamolodchikov. Expectation values of local fields in Bullough-Dodd model and integrable perturbed conformal field theories. *Nucl. Phys. B* **516**, 652–674 (1998).

[81] T. Nishino and K. Okunishi. Numerical latent heat observation of the $q = 5$ Potts model. *J. Phys. Soc. Japan* **67**, 1492–1493 (1998).

[82] R. B. Potts. Some generalized order-disorder transformations. *Proc. Camb. Phil. Soc.* **48**, 106–109 (1952).

[83] A. Kels, Private communication.

[84] R. J. Baxter. A rapidity-independent parameter in the star-triangle relation. arXiv:cond-mat/0108363.

[85] R. J. Baxter, J. H. H. Perk, and H. Au-Yang. New solutions of the star-triangle relations for the chiral Potts-model. *Phys. Lett. A* **128**, 138–142 (1988).

[86] I. G. Enting, A. J. Guttmann, and I. Jensen. Low-temperature series expansions for the spin-1 Ising model. *J. Phys. A* **27**, 6987 (1994).

[87] S. K. Tsang. Square lattice variational approximations applied to the Ising model. *J. Stat. Phys.* **20**, 95–114 (1979).

[88] M. Blume. Theory of first-order magnetic phase change in UO2. *Phys. Rev.* **141**, 517 (1966).

[89] H. W. Capel. On possibility of first-order phase transitions in Ising systems of triplet ions with zero-field splitting. *Physica* **32**, 966 (1966).

[90] B. T. Graham and G. R. Grimmett. Random-cluster representation of the Blume-Capel model. *J. Stat. Phys.* **125**, 283–316 (2006).

[91] M. Clusel, J. Fortin, and V. N. Plechko. Alternative description of the 2D Blume-Capel model using Grassmann algebra. *J. Phys. A* **41**, 405004 (2008).

[92] J. Monroe. Blume-Capel model approximated by a sequence of generalized Husimi trees. *Phys. Rev. E* **65**, 1–5 (2002).

[93] Y. Yüksel, Ü. Aknc, and H. Polat. An introduced effective-field approximation and Monte Carlo study of a spin-1 Blume-Capel model on a square lattice. *Physica Scripta* **79**, 045009 (2009).

[94] R. J. Baxter and P. J. Forrester. A variational approximation for cubic lattice models in statistical-mechanics. *J. Phys. A* **17**, 2675–2685 (1984).

[95] R. J. Baxter and P. J. Forrester. Is the Zamolodchikov model critical?. *J. Phys. A* **18**, 1483–1497 (1985).

[96] T. Nishino and K. Okunishi. A density matrix algorithm for 3D classical models. *J. Phys. Soc. Japan* **67**, 3066–3072 (1998).

[97] K. Okunishi and T. Nishino. Kramers-Wannier approximation for the 3D Ising model. *Prog. Theor. Phys.* **103**, 541–548 (2000).

[98] Intel Fortran compiler manual. `http://software.intel.com/sites/products/documentation/hpc/compilerpro/en-us/fortran/lin/compiler_f/index.htm`.

[99] M. Metcalf and J. Reid, *Fortran 90/95 explained*. Oxford University Press, Oxford, New York, 2002.

[100] Python 2.6 documentation. `http://docs.python.org/release/2.6.7/`.

[101] CSV format. `http://en.wikipedia.org/wiki/Comma-separated_values`.

[102] PBS documentation. `http://www.pbsworks.com/`.

[103] National Computer Infrastructure user guide. `http://nf.nci.org.au/facilities/userguide/`.

[104] IEEE standard for floating-point arithmetic. http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4610933.

[105] Intel Itanium architecture specifications. http://www.intel.com/design/itanium/arch_spec.htm.

[106] GNU Fortran manual. http://gcc.gnu.org/onlinedocs/gcc-4.6.0/gfortran/.

[107] EISPACK library. http://www.netlib.org/eispack/.

[108] Gnu bash manual. http://www.gnu.org/software/bash/manual/.

[109] Intel Fortran linking command helper. http://software.intel.com/en-us/articles/intel-mkl-link-line-advisor/.

[110] HDF5 introduction. http://www.hdfgroup.org/HDF5/doc/H5.intro.html.