Renormalization Multigrid (RMG): Statistically Optimal Renormalization Group Flow and Coarse-to-Fine Monte Carlo Acceleration

Achi Brandt[1] and Dorit Ron[2*]

[1] *Department of Applied Mathematics and Computer Science,*
Weizmann Institute of Science, Rehovot 76100, Israel

[2] *Department of Applied Mathematics and Computer Science,*
Weizmann Institute of Science, Rehovot 76100, Israel
E-mail: dron@wisdom.weizmann.ac.il

(Revised June 2000)

[*] Correspondence should be addressed to D. Ron, Dept. of Applied Mathematics and Computer Science, Weizmann Institute of Science, Rehovot 76100, Israel, Tel:+972-8-9342141, Fax:+972-8-9342945, E-mail: dron@wisdom.weizmann.ac.il

Abstract

New renormalization-group algorithms are developed with adaptive representations of the renormalized system which automatically express only significant interactions. As the amount of statistics grows, more interactions enter, thereby systematically reducing the truncation error. This allows statistically optimal calculation of thermodynamic limits, in the sense that it achieves accuracy ε in just $O(\varepsilon^{-2})$ random number generations. There are practically no finite-size effects and the renormalization transformation can be repeated arbitrarily many times. Consequently, the desired fixed point is obtained and the correlation-length critical exponent ν is extracted. In addition, we introduce a new multiscale coarse-to-fine acceleration method, based on a multigrid-like approach. This general (non-cluster) algorithm generates independent equilibrium configurations without slow down. A particularly simple version of it can be used at criticality. The methods are of great generality; here they are demonstrated on the 2D Ising model.

Introduction

A Monte Carlo (MC) simulation aimed at calculating an average of a certain observable is called "statistically optimal" if it achieves accuracy ε in $O(\sigma^2\varepsilon^{-2})$ random number generations, where σ is the standard deviation of the observable. This is just the same order of complexity as needed to calculate, by statistical sampling, any simple "pointwise" average, such as the frequency of "heads" in coin tossing. Our goal is to attain such an optimal performance in calculating much more complicated averages in statistical physics, including in particular thermodynamic limits, i.e., limits approached by the averages of system observables when the system size tends to infinity.

Two basic factors usually prevent naive Monte Carlo calculations of a thermodynamic limit from being optimal, even when $O(\sigma^2\varepsilon^{-2})$ independent samples are indeed enough to average out their deviations down to $O(\varepsilon)$ accuracy. First, to achieve an $O(\varepsilon)$ approximation to the thermodynamic limit, each sample should be calculated on a system of sufficiently large *volume*, that is, a system whose linear size L grows with ε^{-1}; typically $L \sim \varepsilon^{-\rho}$ for some $\rho > 0$. So in d physical dimensions, the required simulation volume for each sample is $L^d = O(\varepsilon^{-\rho d})$. This factor is called the *volume factor*. The second factor is the *critical slowing down* (CSD), i.e., the increasing number n of MC passes needed (at least at the critical temperature) when L grows in order to produce each new (essentially independent) sample; usually $n \sim L^z$, where z is typically close to 2. As a result of these two factors, the cost of calculating the thermodynamic limit to accuracy ε rises as $O(\sigma^2\varepsilon^{-2-\rho d-\rho z})$.

Cluster algorithms (such as Swendsen-Wang [1] and Wolff [2]) are able to eliminate the CSD factor, i.e., attain $z = 0$ for certain models. For other models they can only partly lower z, or not at all. Moreover, they leave the volume factor intact.

Optimal performance, where both the CSD and the volume factors are eliminated, was first demonstrated in calculating various thermodynamic limits for *Gaussian models* with constant coefficients (and also in calculating the critical temperature of the Ising model) [3]. The main tool was the multigrid cycle, which involves coarse-to-fine acceleration, thus eliminating the CSD, and performs most of the sampling at coarse levels, thus eliminating the volume factor. The technique of inter-level transfer was based (as in classical multigrid) on pre-determined interpolation rules. With increasing sophistication of the multigrid cycling and the interpolation rules, optimal performance has subsequently been accomplished also for massive Gaussian models with variable couplings [4], [5], [6]. For example, it has been shown that the susceptibility of a 2D infinite lattice variable-coupling Gaussian model can be calculated to accuracy ε in less than $20\sigma^2\varepsilon^{-2}$ random number generations, independently of the maximal ratio between strong and weak couplings (unlike the severe extra slowness that large such ratios can inflict on pointwise Monte Carlo).

Efforts to extend these interpolation-based multigrid methods to non-Gaussian models have met with only partial success [6], [7], and have eventually led to the techniques described in this report. These techniques include a couple of interconnected procedures, collectively called the *renormalization multigrid method* (RMG), since they combine ideas previously advanced in both those disciplines.

In this report we will show that the RMG method yields statistically optimal calculations for the 2D Ising model. It will however be clear from the description (and from the discussion in Section 6) that the method is very general. Indeed, it has already preliminarily been applied to the XY model, demonstrating optimal results [7]. Proper modifications of the RMG method are now being introduced to such diverse models as molecular mechanics of macromolecules ([8], Section 14.6 and [9]) and atomistic models of fluids ([8], Section 14.7 and [10]). An analogous method is even being developed for solving *deterministic* sets of equations [11]. Moreover, RMG is applicable even for many systems which are *not* governed by a Hamiltonian.

For simplicity, the new techniques are surveyed here in terms of the 2D Ising model with the majority-rule coarsening, as they were first developed. We begin with a short review of the necessary physical background aimed for the unfamiliar reader. A detailed numerical description for the transition probabilities of the block level is presented next, followed by a comparison to the classical coupling-constants representation. Next it is explained how this approach can be used for the calculation of the fixed point and the correlation-length critical exponent. The Monte Carlo coarse-to-fine multiscale acceleration method is then introduced. Finally, the extension to continuous-state models is briefly discussed.

This article is a revision of [12]. Related publications are [13], [14].

1 Physical Background

We briefly summarize some relevant topics in statistical mechanics including the Ising spin model, the Monte Carlo simulations and the Renormalization group formalism.

1.1 Classical Monte Carlo Method

Consider a two dimensional lattice of size $L \times L$, where each site i is occupied by an Ising spin s_i (which can assume only two values: $+1$ or -1). The spins are related by the nearest neighbor (nn) Hamiltonian

$$\mathcal{H}(S) = -K_{nn}S_{nn} \quad ; \quad S_{nn} = \sum_{<i,j>} s_i s_j \ , \tag{1}$$

where S is a configuration (i.e., a realization of the spins $\{s_i\}$ at all sites), K_{nn} is the associated coupling constant (assumed to absorb $1/(k_B T)$, where T is the temperature and k_B is the Boltzmann constant) and the sum runs over all distinct nearest neighbor pairs $< i, j >$. Periodic boundary conditions are assumed. The probability of a certain configuration S is given by the Boltzmann distribution

$$P(S) \sim e^{-\mathcal{H}(S)} \ . \tag{2}$$

The computational task is to calculate macroscopic physical observables of the system

$$< \mathcal{O} > \ = \sum_S \mathcal{O}(S)P(S) \ , \tag{3}$$

where $\mathcal{O}(S)$ is a property such as the two-point correlation function (at distance d) $\Gamma(d) = L^{-2}\sum_{|i-j|=d} s_i s_j$ (where $|i-j|$ indicates the geometric distance between sites i and j), the magnetization $M = L^{-2}\sum_i s_i$, etc. It is of special interest to calculate the thermodynamic limits (i.e., the limit as $L \to \infty$) of such averages at the critical temperature T_c ($K_{nn} = .4406868$) associated with the magnetic phase transition. The high temperature disordered phase is characterized by zero magnetization ($M = 0$), while as the system is cooled it undergoes a transition to the magnetically ordered phase in which most spins tend to align.

The prime mission of Monte Carlo simulations is to generate a sequence of configurations $S_1, S_2, ..., S_n$, such that each configuration appears a number of times proportional to its probability as given by Eq. (2). This restricts the sampling mainly to the most probable (important) configurations. The evaluation of $< \mathcal{O} >$ is then obtained by a *direct* average over those n configurations: $< \mathcal{O} >= n^{-1}\sum_{i=1}^{n} \mathcal{O}(S_i)$, thus avoiding the enormous summation in Eq. (3). The traditional MC creates the next configuration by changing the spins one by one. Consider a single spin s_i. According to the *Heat Bath* rule, for example, the probability of replacing it by $-s_i$ is given by

$$P(s_i \to -s_i) = \frac{1}{1 + e^{2\Delta\mathcal{H}}} \ , \tag{4}$$

where $\Delta\mathcal{H} = \sum_{j:<i,j>} s_i s_j$, which depends only on the four nearest neighbors of i, the spins at sites marked by 1 in Figure 1.

The accuracy of the calculations depends on the size of the statistical sample n, on the lattice size L and on the correlation length ξ. (ξ is characterized by the fact that two spins separated by a distance small compared with ξ are expected to be found more parallel than anti-parallel.) Near the critical temperature the correlation length diverges as

$$\xi \sim (T - T_c)^{-\nu} \ , \ (as\ long\ as\ L \geq \xi) \tag{5}$$

where ν is the universal correlation-length critical exponent known for this model to be equal to unity. (According to the "universality hypothesis" all other systems of two-dimensional uniaxial magnets share the same ν as well as other critical exponents.) Because ξ diverges at T_c, long heat-bath runs are needed to counter the well known critical slowing down, (i.e., the phenomenon, typical to simulations of critical systems, that when L increases so does the number of MC passes needed to produce a new configuration which is substantially independent of a former one). This problem has been extensively addressed using cluster algorithms (consider [15] and references therein), multigrid methods ([3], [4], [5] and [16]) and more. A different approach for studying systems at the critical temperature is the renormalization group method which is presented next.

1.2 The Renormalization Group Methods

The main difficulty posed by systems near their critical points is the necessity to deal simultaneously with many length-scales. The phenomenological renormalization group (RG) method has been widely used in MC simulations of such systems.

Consider a general action of the form

$$\mathcal{H}(S) = \sum_{\alpha} K_\alpha S_\alpha \ , \tag{6}$$

where each "interaction" S_α is a sum of all spin products of some type, (e.g., a sum of all nearest-neighbor products, denoted by S_{nn} in Eq. (1); a similar sum with *next* nearest neighbors, etc.), each K_α is the corresponding coupling constant and the sum runs over all existing possible interactions on the given grid. The probability of a configuration S is still given by Eq. (2).

The RG transformation $\mathcal{R}$ is defined as the projection of a larger (fine) grid onto a smaller (coarse) grid, consisting of fewer degrees of freedom. In practice, the fine grid is divided into cells (of linear size b). Each cell is then assigned with a single coarse variable according to some prescription. For example, consider the majority-rule with $b = 2$: The coarse grid consists again of a 2D array of Ising spins $\{s'_i\}$, each of which being a "block-spin", i.e., its sign representing the sign of the spin sum in a corresponding 2×2 block of fine-level spins. (The sign of zero is taken to be $+$ or $-$, each with probability 1/2). The coarse grid configuration S', thus produced, is governed by a "renormalized Hamiltonian" $\mathcal{H}' = \mathcal{R}(\mathcal{H})$, which again has the general form

$$\mathcal{H}'(S') = \sum_{\alpha} K'_\alpha S'_\alpha \ , \tag{7}$$

where the interaction S'_α has the same form as S_α, but involving the block-spins $\{s'_i\}$. Conventionally, $\mathcal{R}$ is viewed as operating in the space of the coupling constants: $\{K_\alpha\} \xrightarrow{\mathcal{R}} \{K'_\alpha\}$. If the starting Hamiltonian is critical (i.e., given at T_c), by repeating $\mathcal{R}$ a sufficient number of times, the transformation eventually reaches its attraction point, the critical fixed point $\mathcal{H}^* = \mathcal{R}(\mathcal{H}^*)$.

Any action which is in the vicinity of the critical point $\mathcal{H} = \mathcal{H}^* + \delta\mathcal{H}$ can be expanded in terms of the eigenvectors q_i of the linearized form of $\mathcal{R}$, i.e.,

$$\mathcal{H} = \mathcal{H}^* + \sum_i a_i q_i \ \ , \ \ \mathcal{R}(\mathcal{H}) \approx \mathcal{H}^* + \sum_i a_i \lambda_i q_i \ , \tag{8}$$

where λ_i is the eigenvalue associated with q_i. Thus, the deviation from $\mathcal{H}^*$ either grows or diminishes under renormalization according to whether $|\lambda_i|$ is larger or smaller than unity. In general, there may be few eigenvalues obeying $|\lambda_i| > 1$, each describing a *relevant* direction along which $\mathcal{R}$ flows away from the fixed point. For the Ising model, there is only *one* relevant eigendirection (the temperature-like). Its eigenvalue λ^* gives the rate of flow away from the fixed point and is related to the correlation-length critical exponent ν,

$$\nu = \frac{log\ b}{log\ \lambda^*} \ . \tag{9}$$

Hamiltonians that flow into a particular $\mathcal{H}^*$ are all spanned by the *irrelevant* directions q_i for which $|\lambda_i| < 1$.

In the starting Hamiltonian, on the finest grid, all couplings but the one associated with nearest neighbors vanish, i.e., Eq. (6) boils down to Eq. (1). The coarse grid action $\mathcal{H}'$ has to compensate for all the fine interactions lost while coarsening, and clearly it may need many more interactions than merely the nearest neighbors. More generally, the coarse Hamiltonian should consists of all possible couplings that fit on a given size lattice. Since not all could be included in the simulation, the main question is clearly which couplings are most important. The basic assumption of the renormalization group method is that the renormalized couplings K'_α fall off exponentially with the distance between the spins and with the number of spins in each product in S_α. Under that assumption, the general approach is to take all (or most seemingly important) couplings in a pre-chosen restricted distance of interacting spins, and ignore all other (e.g., longer range) couplings. The fixed errors introduced by taking such a finite number of couplings is referred to as “truncation errors”. Many different methods have been proposed over the years for calculating those coarse couplings associated with the renormalized Hamiltonian. For a brief review consider [17] and references therein. However, to the best of our knowledge, no systematic approach has been developed that would select the couplings according to their significance at a given level of statistical sampling, to roughly match the truncation error with the statistical sampling errors.

An extensively used version of RG is the MC renormalization group (MCRG) [18]. In the MCRG, MC simulations (or, when applicable, some cluster updates) are carried out only with the *original* Hamiltonian, on a grid of some given linear size L. On the produced sequence of configurations a number of successive renormalization blockings is performed, producing corresponding sequences of increasingly smaller configurations of block-spins. The method enables approximate calculation of properties of the RG flow, such as critical exponents, *without* direct knowledge of the renormalized Hamiltonian. However, it still involves the explicit definition of the coarse action. The number of times $\mathcal{R}$ can be applied is limited by L, the starting-lattice size. This may result in lack of convergence (to the fixed point) which is the second source of systematic error in the MCRG calculation (the first being the truncation error). A third source of error is finite-size effects caused by the consistent decrease of the linear size of the configurations being analyzed. For more details, consider for instance [19].

Our present work is aimed at overcoming the above drawbacks. To avoid the finite-size effects and to allow enough renormalization steps we choose, once more, to actually *do* calculate the renormalized Hamiltonian. Indeed, as explained below, we have succeeded in developing a novel numerical method that automatically and systematically constructs the transition probabilities of the block-spin (coarse) level.

2 Systematic Representation of Block-Level Transition Probabilities

We introduce our method first for the simplest case of nearest neighbors Hamiltonian. We will then generalize it to a larger range of interactions. Next, we describe the automatic adaptive approach in which this range grows optimally as function of the invested amount of statistical work. Finally, we present some results, exhibiting the statistical optimality of the method.

2.1 Recovery of Nearest Neighbors Hamiltonian

The numerical method presented here is based on the following, rather simple observation. In order to perform a Monte Carlo simulation over the lattice, one need *not* actually know the explicit structure of the Hamiltonian as given, for example, by Eq. (1). Instead, this formula can be replaced by a table of numbers which gives the *conditional probability* $P^4_+(s_1, s_2, s_3, s_4)$ for a spin to be $+1$ given the values (s_1, s_2, s_3, s_4) of its 4 nearest-neighbor spins (the spins marked by 1 in Figure 1). These conditional probabilities are exactly all one needs for carrying out the MC sweep and generate configurations with the desired Boltzmann weights. Conversely, from a given sequence of configurations in equilibrium, the P^4_+ table can easily be estimated by a simple pointwise scan: For each entry (s_1, s_2, s_3, s_4) the total number of occurrences of this neighborhood, and the number in which the middle spin is $+1$, are counted. The ratio between the latter and the former clearly gives an estimation for $P^4_+(s_1, s_2, s_3, s_4)$. In fact, in this case (of just 4 neighbors), due to the symmetries of the model (flipping, rotating and reflecting), only *two* "equivalence classes" of neighborhoods need to be distinguished: The one in which all 4 spins have the same sign and the one where exactly one of them has an opposite sign. The case for which $s_1 + s_2 + s_3 + s_4 = 0$ is a-priori assigned with $P^4_+(s_1, s_2, s_3, s_4) = 1/2$. All the neighborhoods within the same class must have the same P^4_+, or the same P^4_+ upon flipping, hence only their collective statistics needs be gathered.

This observation can be further used to calculate an estimation for the *coarse* grid Hamiltonian. From an MC simulation of equilibrium on a given fine grid, construct P^4_+ for the corresponding sequence of *coarse* configurations obtained by applying the majority-rule projection $\mathcal{R}$. Then this (coarse) P^4_+ table can be used to simulate the block-spin system and calculate the P^4_+ table for the *next coarse* level (consisting of blocks of block-spins) and so on.

The problem is, of course, that the coarse grid action has a longer range than merely nearest neighbors. Next we explain how the conditional probability tables can be extended to represent more general Hamiltonians.

							.							
							.							
							.							
							8							
						7	5	7						
					6	4	3	4	6					
				7	4	2	1	2	4	7				
.	.	.	8	5	3	1	s_i	1	3	5	8	.	.	.
				7	4	2	1	2	4	7				
					6	4	3	4	6					
						7	5	7						
							8							
							.							
							.							
							.							

Figure 1: The marked 40-spin neighborhood of a spin s_i: Each mark is associated with a layer of all the spins sharing the same distance from s_i.

2.2 Generalization: Larger Neighborhoods

A table of conditional probabilities, similar to P_+^4, can of course be constructed for bigger neighborhoods. For example, P_+^8 is produced by considering the 8-spin neighborhood consisting of the nearest and next-nearest neighbors, correspondingly marked by 1 and 2 in Figure 1. By using the above symmetries, the total number of possible different neighborhoods $2^8 = 256$ in the P_+^8 table is reduced to just 27 equivalence classes (out of which 3 are automatically assigned with the probability 1/2). By considering even more distant spins (those marked by 3), the P_+^{12} table can be constructed (with 314 entries, one for each equivalence class), and so forth.

Clearly, the size of the P_+ tables, thus constructed, grows rapidly. A closer observation would immediately indicate, however, that not all entries have the same importance: Only few are probable, while the rest are rare and contain little statistics. This leads to the following (relatively slowly growing) adaptive structure.

2.3 Adaptive Construction of the P_+^m Tables

The size m of the considered neighborhoods and that of the corresponding P_+^m table should actually depend on the amount of statistics being gathered upon running the MC simulation. If only a small amount of statistics is gathered, only the four nearest neighbors are considered and P_+^4 is constructed. With more statistics, all *eight* closest (nearest and next-nearest) neighbors (marked by 1 and 2 in Figure 1) are considered to construct P_+^8 (consisting of 27 equivalence classes). Since not all appear with similar frequencies, some being much more common than others, it is natural and straightforward to further increase the size of the considered neighborhood *only for the most probable ones.* That is, when the amount of statistics for a particular entry, say in P_+^8, is sufficiently large, that neighborhood is *split*, i.e., statistics is gathered for its "*child neighborhoods*": These are neighborhoods consisting of 12 spins (marked by 1,2 and 3 in Figure 1) with the same inner 8 spins as in the "parent" (split) neighborhood, but with some (or all) combinations of the four *subsequent* neighbors (marked by 3 in Figure 1). Thus, the obtained P_+^{12} table contains information only for the children of the most probable configurations in the P_+^8 table and does not necessarily include all 12-spin neighborhood possibilities.

The general rule is to split a neighborhood when some of its children have enough statistics to make the difference between their P_+ values significant, i.e., larger than their standard deviations. Furthermore, not all offsprings of such a split parent have a separate P_+ entry: Only those children exhibiting a significant change in their P_+ compared with their parent's P_+ are tabled separately, while all others (insignificant in their P_+ deviation, mostly due to lack of enough statistics) are *grouped* (merged) into just one additional equivalence class.

Thus, the P_+^{12} table has a *variable* size, with the number of entries depending on the number of splits (and on the type of the splits, as explained in Section 2.4) occurring from P_+^8. Note that the merging rules are important only for reducing the *size* of the constructed tables. The MC simulation which uses these tables is not affected if the splits are employed less carefully and even if merging is completely avoided, as the effective P_+ value for the *parent* remains the *same* either way.

The splitting process can be repeated: Children with enough statistics (in P_+^{12}) may further be split into grand-children with a larger neighborhood (including also spins marked by 4 in Figure 1 to create P_+^{20}), and so on.

The overall resulting table will have the structure of an unbalanced tree: unequal number of offsprings for different nodes. The tree root connects *all* possible equivalence classes of some small m-spin neighborhood. The most probable nodes (neighborhoods) split into children, the most probable of those further split into grand-children, etc. A schematic possible tree-like structure for 12-spin neighborhoods is shown in Figure 2: The 11 double-framed nodes (i.e., exactly all leaves) are the entries of the associated P_+^{12} table, where by "all others" we refer to the merged offsprings, as explained above.

2.4 Additional Algorithmic Details

For convenience, in order to construct the adaptive table of conditional probabilities we usually use a *pre-run* of length, say, 1/10 of the upcoming actual run. During that pre-run all the decisions concerning splitting are made. Thereafter the structure of the tree is known and remains unchanged. In the current status of the algorithm we haven't fully optimized the process of splitting, further code development is still needed. Nevertheless, we found that the splits (of some particular neighborhood to its subsequent children) should better be executed gradually. That is, all possible children should first be grouped into a small number of "clans". For instance, all those having the same *sum* of all spins in the *outer layer* can be put in the same clan. If the number of spins in the outer layer is k, this sum can assume only $k+1$ different values $(k, k-2, ..., 0, ..., -k)$, and thus splitting into such clans can generate at most $k+1$ offsprings (compared to 2^k different children without this grouping). Only clans

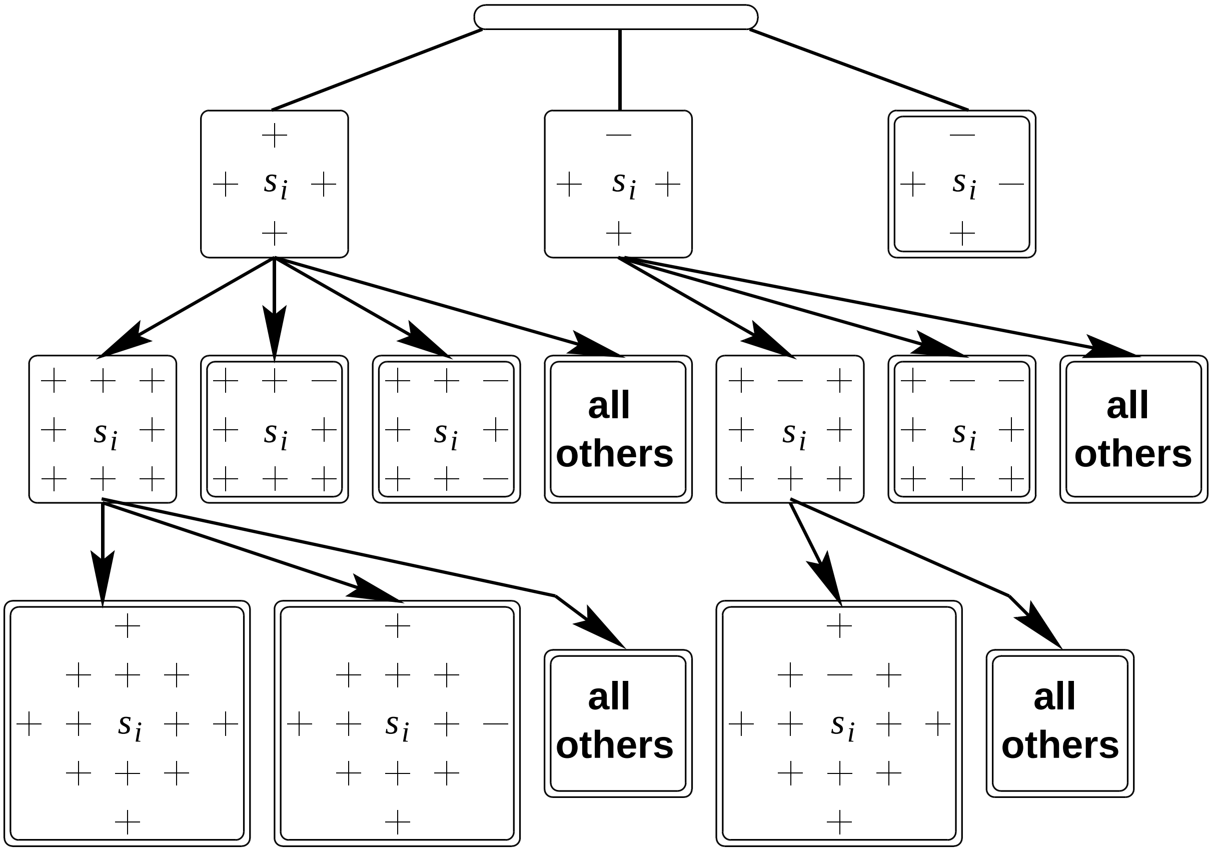

Figure 2: A typical schematic tree-like structure of the P_+^{12} table constructed for all the spins marked by 1,2 and 3. The leaves, i.e., the double-framed nodes, correspond exactly to all the entries of this particular P_+^{12} table.

with enough statistics will further split. With this approach the number of neighborhoods grows at a much slower pace.

In fact, an even better strategy should be the following. For each probable neighborhood, a candidate for splitting, consider at first more than just one possible split, e.g., the split to the next layer's clans, versus the split of the current clan into detailed configurations. Each possible split is then being evaluated, the one with the largest *spread* (average child deviation) should be adopted. (Statistics for *several* possible splits can be accumulated simultaneously in the same MC run.) This would enable a better tuning of the algorithm for achieving full optimality while reducing the number of neighborhoods altogether. This last idea has not yet been implemented.

2.5 Optimal Results in Calculating the P_+ Tables

The P_+ tables represent the coarse-level transition probabilities. Indeed, it is all we need (and exactly what we need) to run an MC simulation on that level (the level of blocks).

Also, due to the adaptability in the size of the neighborhoods, the calculation of the P_+ tables is *statistically optimal*, in the sense that it automatically acquires accuracy ε when the amount of statistics (the total number of random-number generations in producing the P_+ tables) is $O(\varepsilon^{-2})$, without assuming any self-averaging. This claim has been confirmed by the following numerical tests, in which the observables of interest are themselves particular values of the P_+^m table – for the *next*, still coarser level.

From simple heat-bath MC simulations on the finest level we calculate the P_+^m tables of the *first* coarse level, denoting them by $\widetilde{P_+^m}$. From MC simulations on this first coarse level (the level of blocks of spins), using those $\widetilde{P_+^m}$ tables, we then also calculate $\widetilde{\widetilde{P_+^m}}$, the P_+^m tables for the *second* coarse level (the level of blocks of blocks). Note that several different P_+^m tables can be calculated simultaneously in a single MC simulation on a given fine grid. So in the same MC run on the finest grid (using $\mathcal{H}_c$, the original Hamiltonian at T_c), we actually construct *four* different $\widetilde{P_+^m}$ tables: three tables, those with $m = 4, 8$ and 12 (where for $m = 12$ the four spins marked by 3 in Figure 1 are grouped into their 5 possible sums), are used only as the *observables* measured on the first coarse level; the fourth table, which we will denote by $(\widetilde{P_+^m})_{MC}$, is typically constructed for a much larger m (or a more detailed one in case $m = 12$), depending on the length of the MC run (i.e., the amount of statistics). This $(\widetilde{P_+^m})_{MC}$ is the table used for applying the MC simulations on the first coarse level, from which the $\widetilde{\widetilde{P_+^m}}$ tables are measured. Three such tables are actually measured: $\widetilde{\widetilde{P_+^4}}$, $\widetilde{\widetilde{P_+^8}}$ and $\widetilde{\widetilde{P_+^{12}}}$; in this experiment the second-coarse level serves merely for the observable calculations and no MC passes are ever executed on it. At each level, the resulting tables (for $m = 4, 8$ and 12) are then compared with those obtained by a very long Monte Carlo simulation on the *original* grid, using a cluster algorithm (e.g., Wolff). During this original-grid simulations, for each configuration we perform *two* successive majority-rule projections to obtain the first (blocks) as well as the second (block of blocks) coarse configurations, for both of which P_+^m tables (denoted by $\overline{P_+^m}$ and $\overline{\overline{P_+^m}}$, respectively) are measured. The errors we measure are then defined by

$$Error(m,1) = \sum_i |(\widetilde{P_+^m})_i - (\overline{P_+^m})_i| \; \overline{f_i^m} \quad ; \quad Error(m,2) = \sum_i |(\widetilde{\widetilde{P_+^m}})_i - (\overline{\overline{P_+^m}})_i| \; \overline{\overline{f_i^m}} \; , \qquad (10)$$

where i runs over all the entries (in the corresponding table) and $\overline{f_i^m}$ is a non-negative number proportional to the amount of statistics gathered for each $(\overline{P_+^m})_i$, with $\sum_i \overline{f_i^m} = 1$, and $\overline{\overline{f_i^m}}$ is analogously defined.

The results of such calculations on a 32×32 finest grid at the critical temperature T_c ($K_{nn} = .4406868$ in Eq. (1)), are presented in Table 1, where we use the notation $n(L;\mathcal{H})$ to specify the number of MC sweeps employed on an $L \times L$ grid using the Hamiltonian $\mathcal{H}$. Equilibration on the 32×32 grid was first achieved by 1000 MC sweeps starting from a uniform configuration. In all columns (1-7) we present the errors measured for the first and the second coarse levels (averaged over an ensemble of 16 systems). Clearly, in columns 1-4, the errors are halved as the amount of statistics (with correspondingly growing number of neighborhoods) is quadrupled, demonstrating typical optimal behavior. The number of neighborhoods grow faster than it optimally should, since we haven't completely automatized the algorithm with respect to employing minimal number of splits. We have, however, succeeded in reducing the number of neighborhoods much further, as shown in column 6 (compared with column 2), by manually tuning the parameters controlling the necessary splits of the 12-spin neighborhood into its 20-spin-neighborhood children. Also observe (by comparing $Error(m, 2)$ in columns 1-2 with those in columns 5-6, especially for $m = 12$) that all errors should of course be reduced by sufficiently increasing $n(16;(\widetilde{P_+^m})_{MC})$, the amount of MC sweeps (i.e., the statistics) carried on the coarse grid, if one wants to isolate the errors introduced only by the finite statistics $n(32;\mathcal{H}_c)$ used for *producing* $(\widetilde{P_+^m})_{MC}$ and not by the lack of enough statistics in measuring on it the corresponding "observables" $\widetilde{\widetilde{P_+^m}}$. In particular note that for large enough $n(16;(\widetilde{P_+^m})_{MC})$ the $Error(m, 2)$ becomes essentially independent of m.

Finally note that by increasing only the amount of statistics, while keeping the number of neighborhoods in $(\widetilde{P_+^m})_{MC}$ fixed (compare columns 5 and 7), the optimal behavior no longer holds: The errors of the second coarse level are dominated by the truncation error (i.e., truncated neighborhoods) and remain practically unchanged. The required increase in the number of neighborhood is, however, modest: the increase from 304 (column 5) to 461 (column 6) is enough, while the further increase to 1406 (column 2) is no longer helpful.

2.6 Errors

The principal sources of errors in the above processes are the finite statistics, the truncation error imposed by the truncated size of the neighborhoods for which P_+ is calculated, and the finite size of the lattice employed at each level.

The latter type of error is easily removed since arbitrarily large lattices can be used at any coarse level, as the simulation is done directly there, and not through simulations at the finest level. Because of the general numerical form of the P_+ tables, the cluster techniques are inapplicable on those coarse levels. However, they are also unnecessary, first of all due to the *near-locality* (see Section 3) nature of the P_+ calculations at all levels. That is, a very good first approximation for the P_+ tables is already obtained by employing just few simple (heat-bath) MC passes (their number is independent of the lattice size, even starting from a completely random configuration). In particular, it is *not* necessary to obtain *global* equilibrium; it is enough to achieve equilibrium only in a scale comparable to the size of the considered neighborhood. This is evident in Figures 3 and 4, where the errors in calculating P_+^8 and P_+^{12} of the first-coarse-grid (*Error*(8,1) and *Error*(12,1) defined in Eq. (10)), are plotted (on a double logarithmic scale) versus n, the number of MC sweeps employed starting from a random configuration. For small n ($n \leq 10$) results for different gridsizes practically coincide, exhibiting a power law decay (e.g., $Error(8, 1) \propto n^{-1.6}$). For larger n, apparently due to finite-size effects, the rate of convergence for smaller grids is somewhat slower, while results for large enough grids still coincide with each other. Also observe that all results easily exceed the speed needed for optimal behavior, shown by the dashed lines (where the error decreases by a factor of 1/10 when the amount of work increases by a factor of 100).

If a case arises for which faster equilibration and sampling may be needed, they can be achieved by the method of Section 5.

The finite-statistics errors are well controlled so as to keep all of them, at all levels, at

Table 1: The errors in measuring P_+^m of the first and the second coarse levels (for $m = 4, 8$ and 12), using increasingly better approximations and larger neighborhoods for the first coarse level. $\mathcal{H}_c$ is the original Hamiltonian at T_c and the numbers in parenthesis indicate the deviations in the last decimal digits.

	1	2	3	4	5	6	7
$n(32;\mathcal{H}_c)$	10^4	4×10^4	$4^2 \times 10^4$	$4^3 \times 10^4$	10^4	4×10^4	4×10^4
the m in $(\widetilde{P_+^m})_{MC}$	12	20	60	84	12	20	12
number of entries in $(\widetilde{P_+^m})_{MC}$	304	1406	5666	23198	304	461	304
$n(16;(\widetilde{P_+^m})_{MC})$	4×10^4	$4^2 \times 10^4$	$4^3 \times 10^4$	$4^4 \times 10^4$	$4^4 \times 10^4$	$4^4 \times 10^4$	$4^4 \times 10^4$
Error(4,1)	.00050(4)	.00031(4)	.00010(2)	.00005(1)	.00050(3)	.00023(2)	.00022(2)
Error(8,1)	.00064(2)	.00032(1)	.00017(1)	.00008(.4)	.00064(2)	.00032(1)	.00032(1)
Error(12,1)	.00122(2)	.00062(2)	.00030(1)	.00015(.4)	.00122(2)	.00061(1)	.00060(1)
Error(4,2)	.00126(20)	.00061(9)	.00029(6)	.00011(1)	.00134(8)	.00061(4)	.00105(4)
Error(8,2)	.00150(11)	.00074(6)	.00038(3)	.00020(1)	.00140(4)	.00068(3)	.00123(2)
Error(12,2)	.00197(11)	.00101(6)	.00050(3)	.00025(1)	.00149(5)	.00074(3)	.00130(2)

the same optimal order ε, where the amount of statistics is $O(\varepsilon^{-2})$, as was demonstrated above. The truncation errors are also kept at $O(\varepsilon)$, by adjusting the neighborhood sizes; it is estimated that the linear size of the considered neighborhoods should grow very slowly, e.g., proportionately to $\log(\varepsilon^{-1})$. The only remaining trouble is the error enhancement from level to level, due to the renormalization flow divergence away from the critical surface, whose treatment will be discussed in Section 4.2.

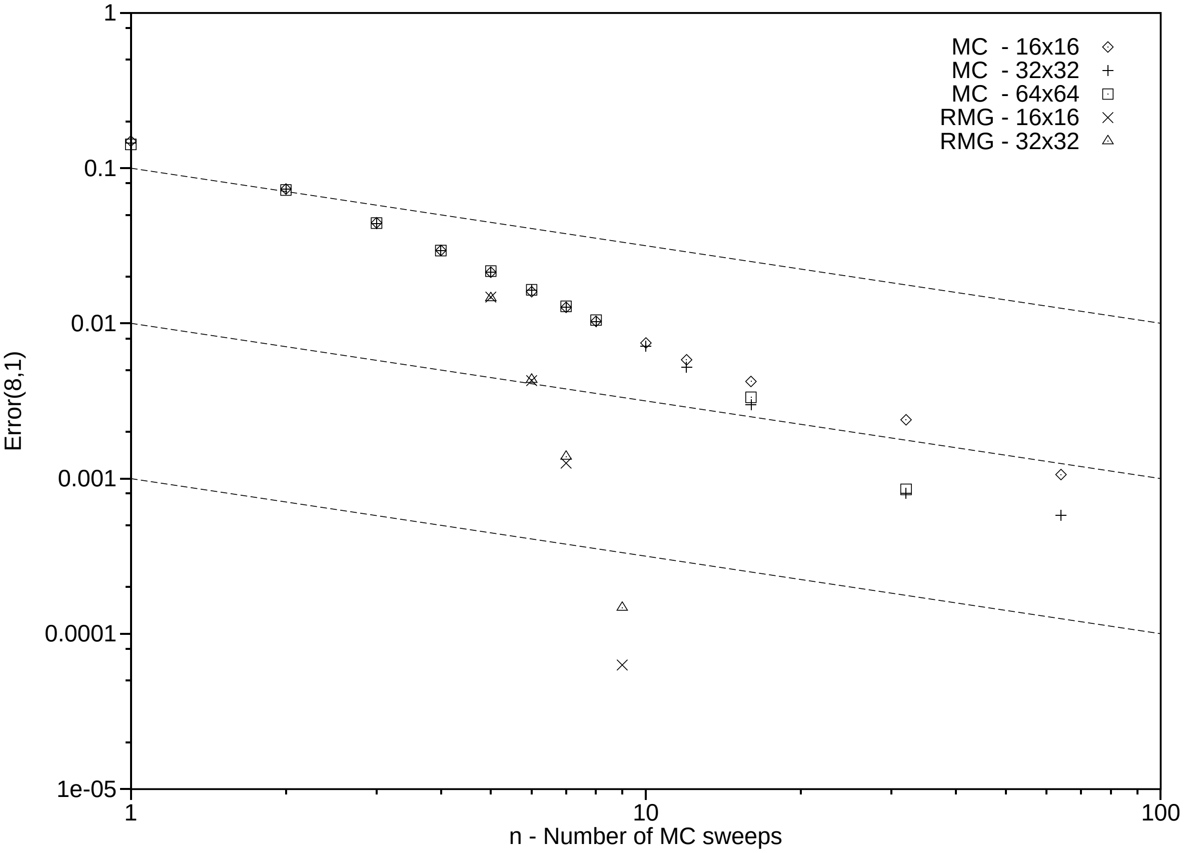

Figure 3: A double logarithmic plot of *Error*(8,1) (defined in Eq. (10)) as a function of n, the number of overall MC sweeps, starting from a random configuration. Results are shown for simple MC simulations (on lattices 16^2, 32^2 and 64^2) and for the TCFE (on 16^2 and 32^2) using the critical Hamiltonian (see Section 5; in this case the CMC and PR passes are included in n).

3 On the Form of Coarse Actions

A general property of coarse (block) levels, in the present model as in most other physical systems, is the *near-locality* of the dependence on neighborhood. That is: the *conditional* probability distribution of the state at a point A, *given fixed states at all other points*, depends mainly on the states of the closest neighbors: the average dependence decays exponentially with the distance from A. (For example, if the neighborhood of A is changed only at points at distances larger than r from A, the conditional probability to have +1 at A can change at most by $O(exp(-cr))$, with some (unknown) constant c.)

[A comment for more general models: The near-locality property of the blocked variables indirectly holds even in the case of long-range interactions, such as electrostatic or gravimetric interactions. Indeed, each such interaction can be decomposed into the sum of a *smooth* part and a *local* part (where "smooth" and "local" are meant relative to the particular scale of the next coarser level). All the smooth parts can be transferred (anterpolated) directly to the coarse level (see [20] and [21]), hence it is only the local parts that remain to be expressed

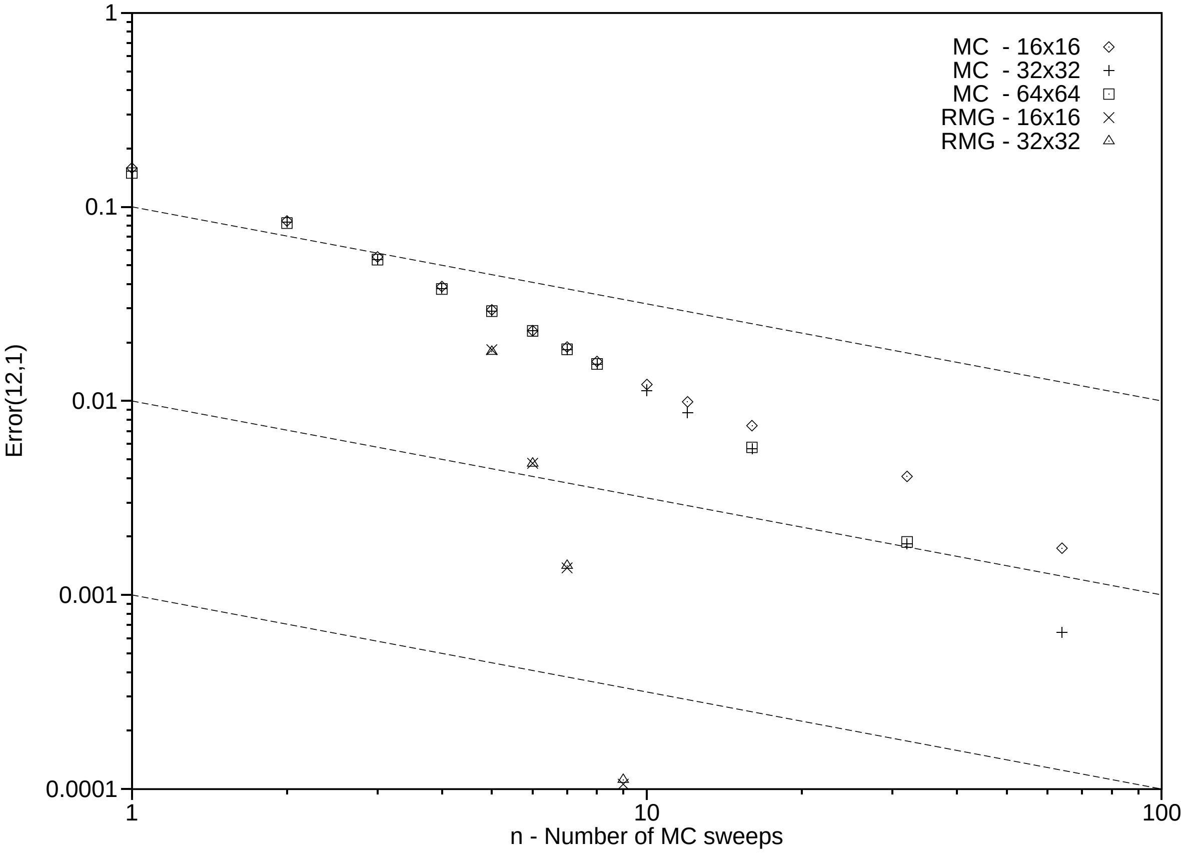

Figure 4: A double logarithmic plot of *Error*(12,1) (defined in Eq. (10)) as a function of n, the number of overall MC sweeps, starting from a random configuration. Results are shown for the same cases as in Figure 3.

on the coarse level. For this expression the near-locality property still holds.]

The near-locality property is of course the motivation behind our approach for the construction of the tree of neighborhoods in terms of which the P_+ table is expressed. It allows us a very systematic branching, which can take far neighbors into account only at the particular combinations and circumstances where their influence is statistically significant. The traditional framework of constructing a coarse-level *Hamiltonian* (by fitting *coupling constants*) does not allow this flexibility. Moreover, it forces one to introduce unnecessarily strong far interactions.

To see this, consider again the P_+^4 table (see Section 2.1), the table related to nearest neighbors. It contains *two* entries, i.e., two conditional probabilities, while there is only *one* nearest-neighbor coupling constant K_{nn} (as in Eq. (1)). Thus, the P_+^4 table can describe a system which is *not* governed by a nearest-neighbor Hamiltonian. To fit such a P_+^4 table, a Hamiltonian description is forced to add farther interactions. Similarly, to fit any P_+^m table the Hamiltonian description is forced to add substantially farther interactions. But by the near-locality property, these farther interactions would generally be much weaker than the P_+^m interactions one attempts to fit.

Note also the fact that the Boltzmann form of probability distributions results from direct physical reasons (like equipartition of energy) which no longer hold for coarse (block) levels (where only a (small) portion of the energy is active). It is thus reasonable to expect that at those levels a more general form of probability distribution will be more suitable.

In particular, our presentation does not suffer from the "peculiarities" of the commonly used discrete-spin RG transformations pointed out by Griffiths and Pearce [22]. As explained in [23], all such pathologies arise form the usual assumption that the renormalization map is defined as a map from Hamiltonians to Hamiltonians. Under this assumption, there are special cases where the renormalized Hamiltonian may fail to exist altogether. Our conditional probability tables have no such restriction. These tables can always be calculated and in the special cases where the renormalized system must include significant longer range interactions, this will be detected by the automatic adaptive construction of the tables and will be taken into account (to the extent that those interactions are indeed significant at the current amount of statistics). This will provide a possible cure for describing those (rare) events.

4 Fixed Point Algorithms and Critical Exponent

As an application of the RMG scheme, we next present two different algorithms which converge to the fixed point of the RG flow. We *assume* that the RG transformation $\mathcal{R}$ has a unique fixed-point (in terms of P_+ table of conditional probabilities) and show that it is obtained in a certain number of iterations (renormalization steps), all carried out with the same lattice size L (unlike the MCRG iterations, where the measurements must be performed on successively decreasing gridsizes). The first method is based on a perturbation (in the relevant direction) introduced into the current approximation to the fixed point, while in the second we employ a "back-to-criticality" mechanism as explained below. Results for the correlation-length critical exponent follow.

4.1 Fixed Point Algorithm by Isolating the Relevant Direction

In the case of the 2D Ising model, the fixed point is quickly approached by a short sequence of coarsening projections (renormalization steps). Let, at a certain stage, the vector P^0 represent the current fine-grid P_+ table of conditional probabilities previously calculated. The RG transformation $\mathcal{R}$ operates on P^0 to produce the coarse-grid P_+ table, which we denote P^1:

$$P^1 = \mathcal{R}[P^0] \ . \tag{11}$$

The vector P^*, which obeys $P^* = \mathcal{R}(P^*)$, is the desired fixed point. As explained in Section 1.2, by applying $\mathcal{R}$ enough times (on a P close enough to P^*), all irrelevant directions diminish, leaving the relevant direction as the dominant perturbation to P. For any perturbation q to a vector P (representing a P_+ table) we define the norm

$$\| q \|^2 \ = \ \sum_i w_i q_i \overline{q_i} \ , \tag{12}$$

where w_i is determined by some rules aimed at minimizing the error in the calculation of λ below (i.e., minimizing the statistical errors in Eq. (14)), from which it follow that $w_i \sim f_i/(P_i(1-P_i))$, with f_i as in Eq. (10) , P_i being the i-th entry in the P vector and $\sum_i w_i = 1$. If the construction of P has been performed very carefully (as described in Sections 2.3 and 2.4) then $\overline{q} = q$. Otherwise, $\overline{q}$ should better be obtained from q by replacing for each parent all entries (q_i) of child neighborhoods which have little statistics, with their weighted average (using w_i for the weighting). This to avoid a possible bias in the norm calculation (Eq. (12)) that may occur due to the contribution of large deviations (whose multiplication each by itself would not average out). A vector q is called normalized if $\| q \| = 1$. Let q^0 be a normalized approximation to q^* (the exact normalized relevant direction) obtained at the previous stage of the algorithm, together with the approximation P^0. Denote by λ^* the eigenvalue associated with q^* (which is in magnitude the largest eigenvalue).

Each *iteration* of the fixed point algorithm combines two parts. In the first part we calculate better approximations for q^* and λ^*. This is achieved by applying $\mathcal{R}$ twice using the same gridsize $(L \times L)$ for both projections:

$$P^1 = \mathcal{R}(P^0) \quad ; \quad P^2 = \mathcal{R}(P^0 + C_q q^0) \ , \tag{13}$$

where $C_q \ll 1$ is a constant, called the perturbation coefficient. At criticality $P^1 = P^0 = P^*$, while $P^2 = P^* + C_q \lambda^* q^* + O(C_q^2)$.

The desired critical exponent can immediately be derived from λ^* by Eq. (9), which in turn is estimated by

$$\lambda = \sum_i w_i q_i^1 \overline{q}_i \Big/ \sum_i w_i C_q q_i^0 \overline{q}_i \ , \tag{14}$$

where $q^1 = P^2 - P^1$. The new approximation to q^* is then defined as $\tilde{q} = q^1 / \| q^1 \|$.

In the second part of the iteration the task is to calculate $\tilde{P}$, a better approximation for the fixed point P^*. We choose

$$\tilde{P} = P^1 + x\lambda\tilde{q} \ , \tag{15}$$

where x is such that $\| \tilde{P} - (P^0 + x\tilde{q}) \|^2$ is minimal. x is thus designed so that $x\tilde{q}$ (nearly) cancels any remaining component in the relevant direction still appearing in P^0.

The next iteration is repeated for $q^0 \leftarrow \tilde{q}$ and $P^0 \leftarrow \tilde{P}$, applied again on the same grid size as the previous iteration.

In principle, the fixed point algorithm should consist of a sequence of *steps*, each consisting of several *iterations* of the (two-part) type defined above. From step to step the amount of statistics should significantly increase, for instance by a factor of 16 (either by employing a growing number n of MC sweeps on a given fixed $L \times L$ grid, or by increasing L, or both), along with the automatized, adaptive increase of the neighborhood's size, supported also by a more accurate equilibration. Also, the perturbation constant C_q should correspondingly be reduced. (Observe that the statistical error involved in the calculations of the P_+ tables is proportional to $L^{-1}n^{-1/2}$, while the error in P^2 is $O(C_q^2)$. Since both errors need to be reduced approximately at the same rate, it follows that upon increasing the amount of statistics by a factor of 16, i.e., decreasing the statistical error by a factor of 1/4, C_q needs to be reduced by a factor of 1/2.) The *first iteration* (in each step) is mainly dedicated for obtaining the new current variables from the former ones. That is, the new P_+ table is constructed for larger neighborhoods (whose choice is based on the neighborhood-frequency information accumulated in the P_+ table of the previous step), and its values are initialized by substituting the parent value into all its new children; the new q vector is similarly initiated and the current initial configuration is simply the last one in the former iteration; then all those values are being updated during the first iteration and serve as (the most updated at this stage) input for the following iterations. These additional iterations *in the same step* (i.e., more iterations each with the same amount of statistics, the same set of neighborhoods and the same C_q) are needed, because the accuracy in calculating λ depends not only on the accuracy of the iteration but also on that of the input (q^0 and P^0).

Thus, step by step, a sequence of systematically improved approximations for the fixed point should in principle be generated, where the overall amount of work is dominated just by the work invested in the very last step.

In practice, each of our λ calculations was conducted mostly with one step and many iterations, all using the same (arbitrarily large) lattice, the same amount of statistics, the same set of neighborhoods and the same C_q. (Previous steps with smaller neighborhoods and much less amount of statistics were used only to obtain first approximations for q^0 and P^0.) We have calculated the average and standard deviation of λ over the ensemble of iterations, discarding the first several of them. Results are given in Section 4.3.

4.2 Fixed Point Algorithm by Repeated Criticalization

In critical calculations, errors introduced at any level are magnified in the level derived from it (the next coarser level), and so on, due to the strong divergence of the renormalization flow away from the critical surface. To check this magnification, a mechanism should be added at each level to project the action P_+ tables back onto the critical surface. Such a "criticalization" mechanism also facilitates calculating renormalization flows toward a fixed point when the critical temperature of the initial (finest-level) Hamiltonian is not known in advance.

The criticalization of a given P_+ table is done by multiplying the temperature by a suitable factor $1/\theta$. In terms of the P_+ tables, this means to raise each probability to the power θ, then normalize; i.e., to replace each value of P_+ by $P_+^\theta/[P_+^\theta + (1-P_+)^\theta]$. The criticalization factor θ can be estimated in a number of ways. In our fixed-point calculations we found it convenient to derive θ from quantities we were calculating anyway, such as the next-level P_+ values and the estimated value of the correlation-length critical exponent (whose derivation is discussed above).

More exactly, in the same MC runs in which P_+ statistics are gathered for the 2×2 block-spins, similar P_+ statistics are also gathered for the 4×4 blocks (meaning, more precisely, 2×2 block of 2×2 blocks), and for the 8×8 blocks, etc. At criticality, these different statistics for different block sizes would coincide (at least for sufficiently large blocks; but close to the fixed point even for small blocks; the use of the latter is preferred since their P_+ tables are based on more statistics and faster equilibration). The differences between P_+ at two different block sizes, together with an (even rough) knowledge of the critical exponent, easily yields an estimate for the needed criticalization factor θ.

This criticalization process may be repeated several times, until those differences between the P_+ values at different block sizes become comparable to the statistical noise. Actually, however, such a repetition is not needed: Applying the process just once at each coarsening step (each renormalization stage) is enough to drive the P_+ table at subsequent levels ever closer to the critical surface. Even better is to apply the criticalization factor directly to the *next* P_+ table (the one that has currently been calculated for the 2×2 blocks). The return-to-criticality cost is then really negligible.

The fixed point of the renormalization group is quickly approached by a sequence of coarsening steps (all implemented successively on the same $L \times L$ gridsize), as described above, with a criticalization factor applied to each new P_+ table in the sequence. Since each iteration should involve a growing amount of statistics (together with an enlarged neighborhood size), the amount of work is, again, dominated by the last iteration. As the fixed point is obtained, the derivation of the eigenvalue λ^* is as described is Section 4.1.

A note on calculating T_c. By observing the P_+ tables over few subsequent renormalization transformations for a given temperature, it is easy to determine whether the temperature is super- or sub-critical. One can therefore trap increasingly narrower intervals around T_c. Provided of course that the P_+ tables become increasingly more accurate (more statistics and correspondingly larger neighborhoods) when narrower intervals are reached. At the same time also increasingly higher levels of renormalization (higher levels of blocking, each with its P_+ table) should be produced. Note however that the algorithm needs only infrequently return to the lower levels, because, to a first approximation around the fixed point, there exists a linear relation between temperature increment at the finest level and increments in the P_+ tables at all coarser levels.

4.3 Results for λ^*

The calculation of λ^*, as given by Eq. (14), has been extensively tested for varying m-spin neighborhoods, values of the constant C_q, amounts of statistics (in calculating P_+^m) and gridsizes. As results from Eq. (14), the standard deviation in the calculation of λ is proportional to $1/C_q$, taking into account that the statistical errors in the calculations of P^1 and P^2 are not related to each other, so they do not cancel out in their difference q^1. The standard deviation marginally grows also as a function of the neighborhood's size, but this may well be due to the imperfection of the current implementation (as mentioned above). Indistinguishable results were obtained for lattices 64^2, 128^2 and 256^2.

Figure 5 shows the resulting λ as a function of the perturbation C_q (in the relevant direction) for 20-spin neighborhood, (consisting of 2826 neighborhoods, where the layer of spins marked by 4 in Figure 1 is considered only via its 9 sums, as explained in Section 2.4) and for 36-spin neighborhood (consisting of $\sim$ 30000 neighborhoods, where all spins marked by 5,6 and 7 are taken via their 17 sums). Each result was averaged over more than hundred iterations (as defined in Section 4.1) so as to guarantee negligible error bars: smaller than .0004 and .0008, respectively. Each of the two $\mathcal{R}$ projections involved in each iteration (see Eq. (13)) was calculated over 10^5 MC passes on a 128^2 grid. The first approximation for the fixed point was obtained from previous steps with less statistics and smaller (8-spin and 12-spin) neighborhoods (consider again Section 4.1).

Since the expansion of P in C_q is linear only near the fixed point, it is clear that if C_q is too large, the perturbation away from the fixed point is too strong and certainly falls off the linear regime. Also, if C_q is too small, the statistical errors, proportional as mentioned to $1/C_q$, violate the calculations. Moreover, we found that even when the amount of statistics grows indefinitely, the results for small C_q fall out of the expected linear dependence on C_q. This is due to the truncation error, and can be explained as follows.

Each neighborhood N_i that has an entry $(P_+)_i$ in our P_+ table can be regarded as the union of "offsprings" (e.g., its "children"): $N_i = \bigcup_j N_{ij}$. Each N_{ij} is a neighborhood coinciding with N_i in its inner layers, and in addition has some specified spin signs in the first layer not included in N_i. It has a frequency w_{ij} and a certain probability, P^+_{ij}, for having a positive spin at its center. Clearly

$$(P_+)_i = \sum_j w'_{ij} P^+_{ij} \ , \tag{16}$$

where $w'_{ij} = w_{ij} / \sum_k w_{ik}$. The perturbation $C_q q$ from the fixed point changes each P^+_{ij} at the next (renormalized) level by $C_q(\lambda q_i + \varepsilon_{ij}) + O(C_q^2)$, where ε_{ij} is small (for large N_i). This contributes $C_q(\lambda q_i + \sum w'_{ij}\varepsilon_{ij}) + O(C_q^2)$ to $(P^2 - P^1)_i$ in our algorithm, which has the desired size (although including the small $O(\varepsilon_{ij})$ error in λ). However, the perturbation $C_q q$ also changes the weights w'_{ij} at the renormalized level, thereby adding an undesired contribution to $P^2 - P^1$, which, by Eq. (16) and the following argument, can be large.

For any fixed neighborhood N_i, the changes in w'_{ij} can mainly be regarded as changes in the expected number of negative (or positive) spins among the spins *just outside* N_i. This number is proportional to the average energy $E =< s_i s_j >$, where s_i and s_j are neighboring spins (at the renormalized level). Hence the changes in w'_{ij} are proportional to the change in E. Since C_q is proportional to a corresponding perturbation $\tau = T - T_c$ in the temperature, the changes of w'_{ij} per unit change of C_q are proportional to derivative of E with respect to T, which is the heat capacity C_p. It is well known that C_p diverges at T_c, hence for C_q tending to 0 (vanishing τ) the changes in w'_{ij} per unit change in C_q will be unbounded, thus introducing *an unbounded error in* λ. This unbounded error results directly from the truncation of neighborhoods; it can be avoided by suitably increasing their size whenever C_q is reduced.

The unbounded truncation error explains the unusual difficulties we have experienced in this particular calculation (computing λ), unlike other RMG calculations. It implies that to achieve higher accuracy in λ one cannot reduce C_q before adequately increasing the neighborhood sizes (as well as the amount of statistics, as mentioned above). Thus, for *fixed* neighborhood sizes, there is a limited range of C_q values for which the computed approximation to λ behaves linearly in C_q.

In Figure 5, the linear regime is clearly shown by the excellent linear fit drawn for the intermediate values of C_q obtained by comparing to other possible (linear) fits over the data and choosing the one which exhibits minimal (least squares per unit length of the C_q interval) error. The resulting estimate for λ^* is obtained by linear extrapolation to $C_q = 0$. For 20-spin

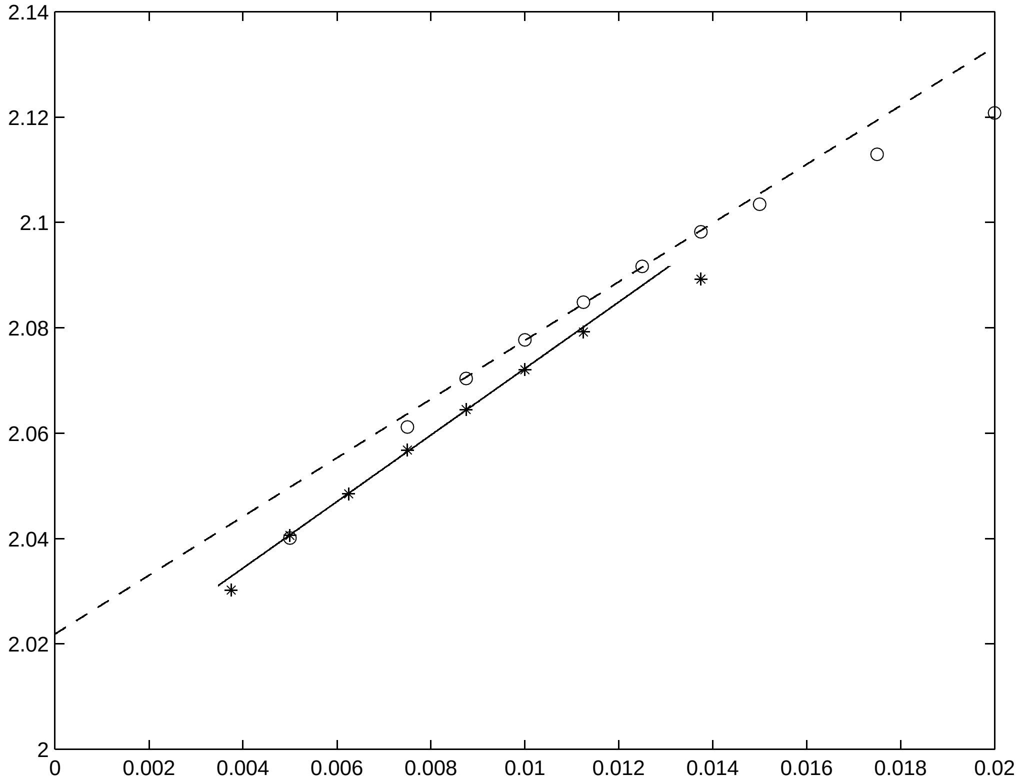

Figure 5: Approximations for λ as a function of the perturbation C_q (in the relevant direction) using P_+^{20} ($\circ$) and P_+^{36} ($*$). The (linear) extrapolated values for $C_q = 0$ are given at the intersections with the y axis.

neighborhood we obtained $\lambda^* \sim 2.022$, for 36-spin neighborhood the improvement was to $\lambda^* \sim 2.009$. This improvement is not so impressive because even in the 36-spin neighborhood we still have the outer layer of the 20-spin neighborhood (marked 4 in Figure 1) taken only in terms of *sums*, which introduces an error not much smaller than that resulting from omitting the next layer (marked 5, 6, 7 in Figure 1). Any of these results can, of course, be improved by increasing the amount of statistics and including more neighborhoods.

5 Coarse-to-fine Monte Carlo Acceleration

For a given lattice with a given action (possibly in the form of P_+ tables), suppose now that the P_+ tables for all its coarser levels (the level of blocks, the level of block of blocks, etc.) are also given. Then a new equilibrium of the given action can accurately and fast be produced using a Monte Carlo coarse-to-fine equilibration (CFE) method, defined as follows.

First an equilibrium is easily obtained at the coarsest level, by few MC passes with the corresponding P_+ table. From this, an equilibrium in the next level is derived, and so on, until the target level (the given lattice) is reached. To obtain an equilibrium in any level of spins given an equilibrium of its blocks, we use "stochastic interpolation", i.e., a number of "compatible Monte Carlo" (CMC) passes. By this we mean Monte Carlo passes at the spin level which keep the values of the blocks unchanged (that is, avoid the processing of every spin whose flipping might change the block variables).

The CMC has a very short autocorrelation time: Actually very close to 1 in all our tests. (More generally, for any model: If (and only if) the CMC autocorrelation time is not short, then the definition chosen for the block variables has been inadequate.) So only few CMC passes are really needed: Their number increases only logarithmically with the desired accuracy; just 4 or 5 of them typically already yield fine results and each additional CMC pass enhances the equilibration by approximately a factor of e (corresponding to autocorrelation time being close to 1). For example, we compared the results for the 2-point correlation function (at distance $\sqrt{2}$) obtained on a 16×16 lattice employing $4, 6$ and 8 CMC passes. The difference between the results with 6 and 8 CMC passes was $\sim .0012$, while the difference between 4 and 8 was $\sim .0092$. The ratio between the two differences being 8.0 which is close to e^2.

If the coarse-level (the block) P_+ table has not been fully accurate, the CMC passes should be followed by a small number of regular MC passes, a process we call "post-relaxation" (PR), following classical multigrid nomenclature. In fact, following again this nomenclature, the above process can be viewed as a "*half-V-cycle*" in which only the second, coarse-to-fine, part of a multigrid V-cycle is employed.

In case of criticality the P_+ tables should have been calculated with criticalization, to avoid the drift away from the critical surface, as explained in Section 4.2.

An extremely simple way to obtain a very good approximation to equilibrium at the critical temperature on a given lattice with a critical action, is by CFE employing *this same action* (e.g., the Hamiltonian given by Eq. (1) at T_c) *at all levels*, with p PR sweeps at each level. We call this process the *trivial* CFE, or TCFE. The produced configurations are completely decorrelated as each one is constructed individually starting from a different small configuration chosen randomly at equilibrium. We have measured 0 autocorrelation time for various observables on up to 256^2 lattices. The cost of a new independent configuration depends on the employed number of CMC passes and on p, but as explained below is independent of the lattice size. It can be shown that the required number p of PR sweeps is small whenever the convergence to a fixed point of the renormalization flow is fast. If the Hamiltonian used is fairly close to the fixed point (i.e., a good approximation for the fixed point is obtained in just few RG steps), then using it on a particular grid produces nearly the correct equilibrium for block-spins of somewhat coarser levels. Thus, the PR is needed mainly to equilibrate only the smaller, *local* scales. This is indeed evident in the following numerical results, which exhibit the excellent quality of equilibria obtained by TCFE.

In Table 2 we present the errors measured for the two-point correlation function (at distance $\sqrt{2}$ meshsizes). The errors are calculated by comparing to results obtained from long runs of the Wolff algorithm. The table shows that the errors are fixed as the lattice grow and decrease rapidly with the number of post-relaxations *independently of the lattice size.* Similar results were obtained for other observables (e.g., the energy).

Also, as shown in Figures 3 and 4, very small errors were measured in P_+^8 and P_+^{12} of the first-coarse-level over an ensemble of configurations produced by the TCFE (with 4 CMC

Table 2: The errors in measuring the two-point correlation function (at distance $\sqrt{2}$ meshsizes) by TCFE, with 8 CMC passes and p post-relaxation sweeps, on an $L \times L$ grid.

p	0	4	8	16
$L = 16$	.00513	.00068	.00044	.00017
$L = 32$	.00682	.00139	.00079	.00039
$L = 64$	.00772	.00174	.00094	.00049
$L = 128$	.00825	.00233	.00100	.00049
$L = 256$	.00851	.00215	.00095	.00046
$L = 512$	.00869	.00210	.00109	.00059

sweeps) starting from a completely random configuration at the coarsest level. The errors were again calculated by comparing figures with those of the Wolff algorithm. Results are shown for $p = 0, 1, 2$ and 4 PR sweeps for grids 16^2 and 32^2, where the amount of work taken into account includes the 4 CMC sweeps, the p PR passes and an additional one which roughly stands for the work accumulated on all coarser levels of the TCFE. Thus, the four results are drawn versus $5, 6, 7$ and 9 MC sweeps, respectively. Note the much accelerated pace of convergence per MC pass brought about by the TCFE. Also observe that measurements for the larger neighborhood (e.g., of 12-spins) is less sensitive to finite-size effects than smaller neighborhood (of 8-spins), hence the former exhibits a more regular behavior than the latter.

Remember that these results refer only to the *trivial* CFE. They can be improved by using on all coarse levels not the finest grid critical Hamiltonian $\mathcal{H}$ but the Hamiltonian $\mathcal{R}(\mathcal{H})$ (in the form of P_+ tables). For example, we have calculated P_+^{12} and P_+^{20} of the first-coarse-level over 4×10^6 MC sweeps on an 128^2 lattice at T_c. As shown in Table 3 the accuracy in calculating the two-point correlation function is significantly improved (compare with Table 2) by using these P_+ tables on all *coarse* levels (even with $p = 0$ PR sweeps). Still further improvement can presumably be obtained by using $\mathcal{H}$ at the finest level, $\mathcal{R}(\mathcal{H})$ on the next coarser level and $\mathcal{R}^2(\mathcal{H})$ on all other levels; etc, provided each additional projection $\mathcal{R}$ involves a proper criticalization. Without such criticalization, on sufficiently coarse level it is better to use the original Hamiltonian $\mathcal{H}$ (if it is known to be critical).

Table 3: The errors in measuring the two-point correlation function (at distance $\sqrt{2}$ mesh-sizes) by CFE using P_+^{12} and P_+^{20} on all coarse levels with 8 CMC passes and $p = 0$ on an $L \times L$ grid.

	12-spin neighborhood	20-spin neighborhood
$L = 64$	.00143	.00076
$L = 128$	.00135	.00089

6 Extension to Continuous-State Models

Initial steps of applying the above coarsening and acceleration techniques to the XY model are reported in [7]. Each 2×2 block spin is here defined to be the *average* of its four constituent spins, *without normalization* (whereby the original XY group of length-1 vectors is *not* preserved at the coarse levels). Compared with the ± 1 majority spins discussed above, each coarse spin here contains much more information; as a result, much smaller neighborhoods are needed in the probability tables to attain a given truncation accuracy. Still, these tables are more complicated than the above P_+ tables, since they should describe a *continuous* distribution, conditioned on *continuous* neighboring values.

To accumulate continuous-variable statistics, one of course partitions the range of this variable into *bins*: Counting the number of MC hits in each bin gives an estimate for the *integral* of the continuous variable over that bin. From those integrals, the value of the variable at any particular point can be *interpolated* (by a polynomial each of whose integrals over several adjacent bins fits the estimate). The same is true for a *vectorial* variable, such as the one representing the entire (truncated) neighborhood, whose bins may each be a tensor product of *elementary* bins, where each elementary bin is one of the bins of one of the real variables making up the vector. More generally, the bins of the neighborhood are constructed *adaptively*, similar to the adaptive neighborhoods in the Ising case above, except that here a bin can be split into several bins in *two* ways: either by adding another variable to the description of that particular neighborhood, or by refining the current bin partition of one of the existing variables.

The set of tests with the XY model reported in [7], though still limited to the simplest neighborhood, clearly indicates that ideal MC performance is obtained in calculating various thermodynamic limits, such as the two-point correlation and the scaled susceptibility.

After further program improvements (more accurate and automatic implementation of the rules described above) and further testing of optimality for various observables, we plan to extend the RMG techniques to more advanced physical problems, including gauge field models such as U(1), SU(2) and SU(3).

Acknowledgments

We would like to thank R. H. Swendsen for his enlightening remarks.
The research has been supported by grants No. G0289-065.07/93 from The German-Israeli Foundation for research and development (GIF) and No. 696/97 from the Israel Science Foundation, by AFOSR contract F33615-97-D5405 and by the Carl F. Gauss Minerva Center for Scientific Computation at the Weizmann Institute of Science.

References

[1] R. H. Swendsen and J. S. Wang. Nonuniversal critical dynamics in Monte Carlo simulations. *Phys. Rev. Lett.*, 58:86–88, 1987.

[2] U. Wolff. Collective Monte Carlo updating for spin systems. *Phys. Rev. Lett.*, 62:361–364, 1989.

[3] A. Brandt, M. Galun, and D. Ron. Optimal multigrid algorithms for calculating thermodynamic limits. *J. Stat. Phys.*, 74:313–348, 1994.

[4] A. Brandt and M. Galun. Optimal multigrid algorithms for the massive Gaussian model and path integrals. *J. Stat. Phys.*, 82:1503–1518, 1996.

[5] A. Brandt and M. Galun. Optimal multigrid algorithms for variable-coupling isotropic Gaussian models. *J. Stat. Phys.*, 88:637–664, 1997.

[6] M. Galun. Statistically Optimal multigrid algorithms in statistical physics. Ph. D. Thesis, Weizmann Institute of Science (1998).

[7] S. Shmulyian. Towards optimal multigrid Monte Carlo computations in two-dimensional $O(N)$ non-linear σ-models. Ph. D. Thesis, Weizmann Institute of Science (1999).

[8] A. Brandt. Multiscale Scientific Computation: Six year Research Summary. In the web site http://www.wisdom.weizmann.ac.il/∼achi.

[9] D. Bai and A. Brandt. Multiscale Computation of Polymer Models. *This Proceedings.*

[10] A. Brandt and V. Ilyin. Multiscale Approach in Statistical Phisics of Liquids. *This Proceedings.*

[11] A. Brandt. General Highly Accurate Algebraic Coarsening Schemes. *Electronic Trans. Num. Anal.*, 10:1–20, 2000.

[12] A. Brandt and D. Ron. Statistically Optimal Renormalization Group Flow and Coarse-to-Fine Monte Carlo Acceleretion. Gauss Center Report WI/GC-11, Weizmann Institute of Science (1999).

[13] A. Brandt and D. Ron. Renormalization Multigrid (RMG): Action Perfection and Slow-Down Elimination. Submitted to Phys. Rev. Lett., July 1999.

[14] A. Brandt and D. Ron. Renormalization Multigrid (RMG): Statistically Optimal Renormalization Group Flow and Coarse-to-Fine Monte Carlo Acceleretion. Submitted to J. Stat. Phys, July 1999.

[15] W. Janke. Monte Carlo simulations of spin systems. In *Computational Physics*, pages 10–43, Berlin, 1996. Springer. Hoffman K. H. and Schreiber M. (Eds.).

[16] A. Brandt and M. Galun. Statistically Optimal multigrid algorithms for the anharmonic crystal model. Gauss Center Report WI/GC-9, Weizmann Institute of Science (1998).

[17] R. Gupta. Open problems in Monte Carlo renormalization group: Application to critical phenomena. *J. Appl. Phys.*, 61:3605–3611, 1987.

[18] R. H. Swendsen. Monte Carlo Renormalization. In T. W. Burkhardt and J. M. J. van Leeuwen, editors, *Real-Space Renormalization*, pages 57–86. Springer, Berlin, 1982.

[19] C. F. Baillie, R. Gupta, K. A. Hawick, and G. S. Pawley. Monte Carlo renormalization-group study of the three-dimensional Ising model. *Phys. Rev. B*, 45:10438–10453, 1992.

[20] A. Brandt. Multilevel computations of integral transforms and particle interactions with oscillatory kernels. *Comp. Phys. Comm.*, 65:24–38, 1991.

[21] B. Sandak and A. Brandt. Multiscale Fast Summation of Long Range Charge and Dipolar Interactions. *This Proceedings.*

[22] R. B. Griffiths and P. A. Pearce. Mathematical Properties of Position-Space Renormalization-Group Transformations. *J. Stat. Phys.*, 20:499–545, 1979.

[23] A. C. D. van Enter, R. Fernandez, and A. D. Sokal. Regularity Properties and Pathologies of Position-Space Renormalization-Group Transformations: Scope and Limititions of Gibbsian Theory. *J. Stat. Phys.*, 72:879–1167, 1993.