

New method for the Hamiltonian formulation for lattice spin systems

H. H. Roomany, H. W. Wyld, and L. E. Holloway

Department of Physics, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801

(Received 3 December 1979)

The Lanczos scheme for finding low-lying eigenvalues of a sparse matrix of large dimension is applied to solving the Hamiltonian formulation for $Z(2)$ and $Z(3)$ Ising spin systems in $1+1$ dimensions. The location of the fixed points and the values of the critical indices are consistent with known results.

I. INTRODUCTION

In this paper we study the Hamiltonian formulation^{1,2} for $Z(2)$ and $Z(3)$ Ising spin systems in $1+1$ dimensions, looking toward future application to lattice Hamiltonian field theories.³

For such systems two methods have been used. One is the Hamiltonian analog⁴ of the high-temperature expansion employed in statistical mechanics. In this method one uses perturbation theory to generate power-series expansions for the energy levels to rather high order in the coupling. These series are then extrapolated to the large-coupling region by Padé methods. Critical properties, masses of excitations, and other quantities of interest are obtained from the Padé approximants. The other method is the real-space renormalization-group transformation.^{5,6} Here, one would examine how certain physical quantities change as the length scale increases.

A third method has recently been proposed by Hamer and Barber.⁷ Their method consists of comparing a sequence of finite lattices. The finite-lattice systems are solved exactly, and various quantities can be calculated as functions of the lattice size L , for small L . Finally, these functions are scaled up to $L=\infty$. Two steps are needed before these ideas can be realized. First, one needs a procedure for solving the finite-lattice system exactly. This involves diagonalizing real symmetric matrices of enormous dimensions. Second, one needs a procedure for extrapolating from finite to infinite L . This extrapolation procedure has already been found and used⁸ in statistical mechanics. In this paper we present a method for handling step one above. We believe that our method is better than that used in Ref. 7, for reasons given in Sec. II.

This paper is organized as follows: In Sec. II the method is explained. In Sec. III we apply the method to the $Z(2)$ and $Z(3)$ models in $1+1$ dimensions. We compare our results with those obtained by the other two methods (perturbation theory and renormalization group) in Sec. IV.

II. THE METHOD

The problem can be formulated in general as follows. Given a Hamiltonian defined on a finite lattice having N sites with n states each, find the low-lying eigenvalues of H . The standard method of implementing this on a computer would involve calculating a matrix representation of H (an $n^N \times n^N$ matrix) and then using standard computer routines to first transform this matrix into tridiagonal form, then diagonalize it. This method is, unfortunately, impractical for our purposes because even for the simplest model [$Z(2)$ in one spatial dimension] one has $n=2$ and a $2^N \times 2^N$ matrix is too large to be stored in the computer's central memory for $N > 8$.

An alternate method due to Lanczos⁹ uses a special basis so that the matrix representation of H is immediately tridiagonal. That means, instead of storing $n^N \times n^N$ numbers one has to store $2n^N$ of them only. This method has been extensively used in shell-model calculations in nuclear physics.⁹ The basis can be constructed as follows. Let ψ_1 be an arbitrary, normalized state of the system. ψ_1 is the first element of the basis. The remaining elements can be generated recursively from ψ_1 by repeated applications of H :

$$H\psi_1 = a_1\psi_1 + b_1\psi_2.$$

Here, a_1 and b_1 are chosen so that ψ_2 is normalized and orthogonal to ψ_1 . We act with H again,

$$H\psi_2 = c_1\psi_1 + a_2\psi_2 + b_2\psi_3,$$

where again one chooses c_1 , a_2 , and b_2 so that ψ_3 is normalized and orthogonal to both ψ_1 and ψ_2 . We continue along these lines: each time a new state ψ_n is generated, we compute the state $H\psi_n$, subtract off its components along $\psi_1, \psi_2, \dots, \psi_n$ and normalize what is left over, which is ψ_{n+1} ,

$$H\psi_n = \sum_{i=1}^{n-2} \alpha_i \psi_i + c_{n-1} \psi_{n-1} + a_n \psi_n + b_n \psi_{n+1}. \quad (1)$$

(We have separated off the first $n-2$ components for reasons to become clear shortly.) The $n+1$

states $\psi_1, \psi_2, \dots, \psi_{n+1}$ generated thus far are, by construction, orthonormal. If we take the projection of Eq. (1) on the state ψ_j ($j \leq n-2$), we find

$$(\psi_j, H\psi_n) = \alpha_j.$$

We use the fact that H is Hermitian to rewrite this as

$$(H\psi_j, \psi_n) = \alpha_j.$$

However, we see from Eq. (1), applied with $n=j$, that $H\psi_j$ is a linear combination of states, each of which is orthogonal to ψ_n . Hence,

$$\alpha_j = 0, \quad j = 1, 2, \dots, n-2.$$

Thus we have the following structure:

$$H\psi_1 = a_1\psi_1 + b_1\psi_2,$$

$$H\psi_2 = c_1\psi_1 + a_2\psi_2 + b_2\psi_3,$$

$$H\psi_n = c_{n-1}\psi_{n-1} + a_n\psi_n + b_n\psi_{n+1}.$$

The Hermiticity of H also implies $b_i = c_i$, $i = 1, 2, \dots$. In the basis ψ_1, ψ_2, \dots , H has the tridiagonal form

$$\begin{pmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & b_2 & a_3 & b_3 & \\ & & \ddots & \ddots & \ddots \\ & & & & & \ddots \end{pmatrix}.$$

The Lanczos process terminates automatically when the states ψ_1, ψ_2, \dots span the entire space of H , or a sector of it connected by nonvanishing matrix elements of H .

The most attractive feature of the method is that one does not have to carry the process to completion in order to have accurate estimates of the low-lying eigenvalues. The low-lying eigenvalues of the leading $p \times p$ principal submatrix of H , which is completely determined after $p-1$ steps, converge extremely rapidly to those of H . One can monitor this convergence numerically by diagonalizing the tridiagonal matrix after each step and watching how the eigenvalues change. We have found for the cases we studied that with $p=20$, that is, after 20 Lanczos steps, the lowest eigenvalue was accurate to machine accuracy, irrespective of the size of n or N . Detailed evidence for the rapid convergence in the shell-model problem is presented in Ref. 9.

The number of states involved is further reduced because of the fact that we study only translationally invariant states. (We use periodic boundary conditions throughout.) Thus, instead of having to deal with n^N states, one needs only certain linear combinations of them. In the $Z(2)$ case, for ex-

ample, if one starts with a spin configuration ψ_1 having all N spins down, then only states with an even number of up spins can be generated because the potential-energy piece of H flips two spins at a time. The number of translationally invariant states with an even number of up spins is $M = (2^{N-1} + N - 1)/N$ for N a prime number and, hence, the Lanczos process will terminate after $M-1$ steps. The last Lanczos state will be a linear combination of M translationally invariant states. Special techniques had to be used to store and manipulate such many-component states. These techniques are explained in Appendix B.

In applying the method, one starts off with the weak-coupling ground state ψ_1^0 , obtained by putting each site in its ground state. The lowest eigenvalue generated from this ψ_1^0 is the ground state E_0 of the system. To get E_1 , the first excited state of the system, one starts with the weak-coupling first excited state ψ_1^1 , having all sites in their ground states except one, which is in its first excited state. From the results of these calculations one obtains the free energy $F = E_0/N$, and the mass gap $G = E_1 - E_0$. Note that the validity of the method is independent of the size of the coupling constant x in H . Thus, the functions $F(x, N)$ and $G(x, N)$ can be calculated *exactly* at any x . Scaling arguments are then used to extract information from F and G about the $N \rightarrow \infty$ limit. These arguments are discussed and used in the next section. We also investigated the possibility of obtaining F and G of the infinite lattice directly, without using scaling, by simulating an infinite lattice on a finite one. The results obtained this way are rather interesting and are discussed in Appendix A.

The method used in Ref. 7 is similar to ours in that it involves generating a special basis and then diagonalizing the matrix representation of H in that basis. Unlike our method, however, their method does not render the matrix representation of the complete Hamiltonian H tridiagonal. Specifically, they generate a basis ϕ_0, ϕ_1, \dots by starting with an arbitrary normalized spin configuration ϕ_0 and then form the states $V^n \phi_0$ ($n = 1, 2, \dots$), where V is the potential-energy part of the Hamiltonian $H = K + V$. Note that since V is a sum of N terms, each time one operates with V , one generates N states (some of which are translations of each other). The generation process terminates for $Z(2)$, after $Q = (N+1)/2$ steps, for N odd. This way NQ states will be generated of which L (say) are translationally invariant. The states $\phi_0, \phi_1, \phi_2, \dots, \phi_L$ are decoupled from the rest of the states and, hence, H can be diagonalized separately in this L -dimensional space. The disadvantage of this scheme relative to ours is that the resulting L -dimensional repre-

sentation of H is not tridiagonal and, thus, the scheme is limited, practically, to cases with $L \lesssim 200$.

III. APPLICATIONS

A. The fixed point

The fixed point of an infinite system is defined, in the Hamiltonian formulation, as the value of x , x^* , at which the mass gap $G(x)$ vanishes. For Ising-type systems, $G(x)$ is expected² to have an algebraic zero at x^* :

$$G(x) \sim G_0(x^* - x)^\nu, \text{ as } x \rightarrow x^* \quad (2)$$

where ν is the correlation length index: $\xi \rightarrow \xi_0(x^* - x)^{-\nu}$ as $x \rightarrow x^*$. With our Lanczos scheme we can compute $G(x, N)$, which approaches $G(x)$ when N is large. Figure 1 shows $G(x, N)$ as a function of x for several N values for the $Z(2)$ model whose Hamiltonian is

$$W = \sum_{i=1}^N \{ [1 + \sigma_x(i)] - x \sigma_x(i) \sigma_x(i+1) \}.$$

Also shown in Fig. 1 is $G(x) = G(x, \infty)$. Figure 2 is the corresponding plot for the $Z(3)$ model whose Hamiltonian is¹⁰

$$W = \sum_{i=1}^N \left[\frac{2}{3} \left(1 - \cos \frac{2\pi l_i}{3} \right) - x \cos \left(\frac{2\pi}{3} (m_{i+1} - m_i) \right) \right],$$

$$[l_i, m_j] = -i \frac{3}{2\pi} \delta_{ij}.$$

We see from these two figures that $G(x, N)$ does not vanish at any finite value of x in agreement

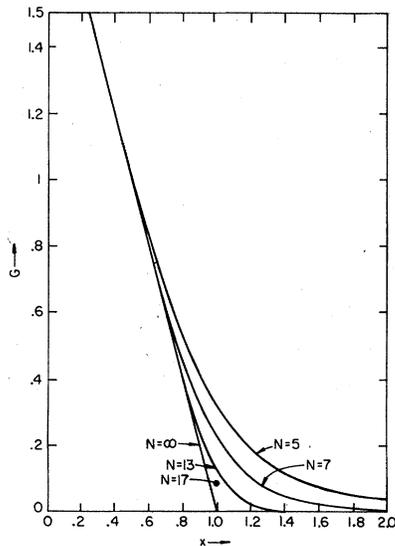


FIG. 1. The mass gap $G(x, N)$ of the $Z(2)$ model as a function of x for several N values. The straight line is the exact mass gap of the infinite system ($N = \infty$).

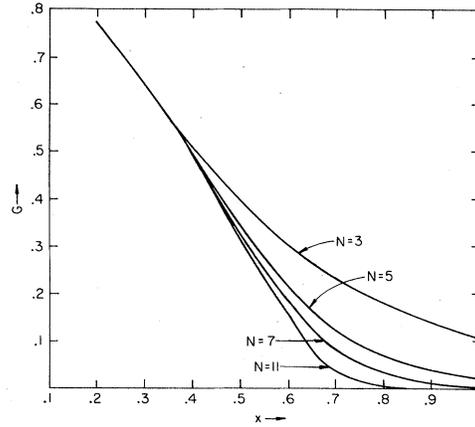


FIG. 2. Similar to Fig. 1, but for $Z(3)$.

with the fact that a finite system cannot have a phase transition.

One can, nevertheless, extract x^* from $G(x, N)$ by the following argument,^{7,8} which we reproduce for the convenience of the reader. The natural measure of the deviation of the finite system from the infinite one is L/L_0 , where L is the linear dimension of the finite system ($L = Na$, a is the lattice spacing) and L_0 is the correlation length of the infinite system ($L_0 = \xi a$). Thus, it is natural to assume that $G(x, N)$ depends on x through L/L_0 and we write, for the x dependence of $G(x, N)$,

$$G(x, N) \propto f(L/L_0) = f(N(x^* - x)^\nu / \xi_0).$$

If we are to retrieve Eq. (2) as $N \rightarrow \infty$, we have to assume that the asymptotic functional form of f is a power

$$f(y) \sim f_0 y^p \text{ as } y \rightarrow \infty.$$

In addition, we need a factor to cancel the N dependence of f as $N \rightarrow \infty$. This factor must be of the form N^{-p} . Thus, we have

$$G(x, N) = N^{-p} f(N(x^* - x)^\nu / \xi_0). \quad (3)$$

p can be easily determined by equating $G(x, N)$ of Eq. (3) to $G(x)$ of Eq. (2) in the limit $N \rightarrow \infty$, $x \rightarrow x^*$:

$$N^{-p} f_0 (N(x^* - x)^\nu / \xi_0)^p = G_0 (x^* - x)^\nu,$$

which implies

$$f_0 = \xi_0^p G_0 \text{ and } p = 1.$$

So Eq. (3) becomes

$$G(x, N) = N^{-1} f(N(x^* - x)^\nu / \xi_0)$$

or

$$\ln G(x, N) = \ln f(N(x^* - x)^\nu / \xi_0) - \ln N. \quad (4)$$

Equation (4) states that if we plot $G(x, N)$ against N on a log-log plot, we will get a straight line of

slope -1 only if $x = x^*$ and provided that N is large enough for the scaling of Eq. (4) to set in. Such a plot is shown in Fig. 3 for the $Z(2)$ model and in Fig. 4 for the $Z(3)$ model. We see in this figure that for $x < 1$ ($x > 1$), $G(x, N)$ tends to curve up (down) away from the slope $= -1$ direction. Near $x = 1$ is a transition region between these two regimes and it is in this region where one should look for x^* .

Since one does not know, *a priori*, the value of N at which scaling sets in, one has to consider several N values and examine the results. We used the following procedure. We evaluate $G(x, N)$ for three values of N , N_1 , N_2 , and N_3 , chosen as large as possible. The slope of $G(x, N)$ between N_1 and N_2 , on a log-log plot against N , is given by

$$S_{N_1, N_2}(x) = \ln[G(x, N_1)/G(x, N_2)] / \ln(N_1/N_2).$$

We write

$$S_{N_1, N_2}(x) = -1 + d_1(x),$$

$$S_{N_2, N_3}(x) = -1 + d_2(x),$$

and take x^* to be $\bar{x} \pm e$, where \bar{x} is a value of x that minimizes, as well as possible, both $|d_2(x)|$ and $|S_{N_1, N_2}(x) - S_{N_2, N_3}(x)|$, and e is the larger of the two numbers $|d_2(\bar{x})|$ and $|S_{N_1, N_2}(\bar{x}) - S_{N_2, N_3}(\bar{x})|$.

We calculated $G(x, N)$ of $Z(2)$ for $N_1 = 9$, $N_2 = 11$, and $N_3 = 13$ and fitted the results to polynomials for x close to x^* :

$$G(x, 9) = 3.453 - 6.044x + 3.166x^2 - 0.400x^3,$$

$$G(x, 11) = 3.031 - 4.394x + 1.053x^2 + 0.453x^3,$$

$$G(x, 13) = 3.953 - 6.779x + 2.999x^2 - 0.053x^3.$$

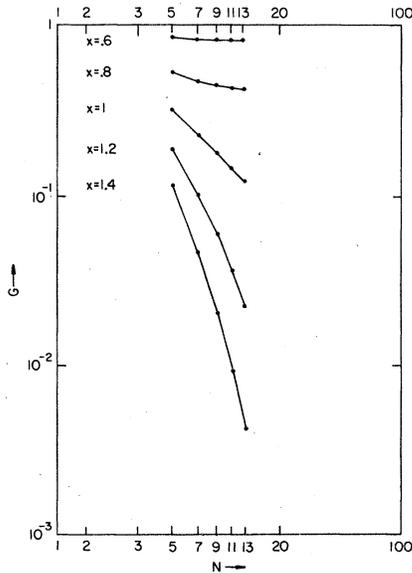


FIG. 3. Log-log plot of $G(x, N)$, of the $Z(2)$ model, versus N for several x values.

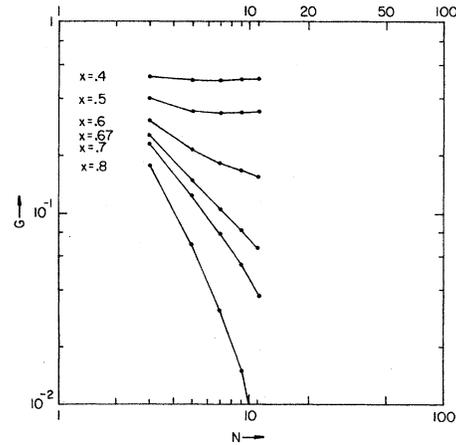


FIG. 4. Similar to Fig. 3, but for $Z(3)$.

Using these fits to carry out the procedure outlined above, we obtained for the critical point x^* the value 0.9995 ± 0.0010 .

The same type of analysis was done for the $Z(3)$ model with $N_1 = 7$, $N_2 = 9$, and $N_3 = 11$ (see Fig. 4). The fit for x close to x^* was

$$G(x, 7) = 2.202 - 5.359x + 3.321x^2,$$

$$G(x, 9) = 2.862 - 7.346x + 4.763x^2,$$

$$G(x, 11) = 3.635 - 9.659x + 6.460x^2,$$

yielding for $Z(3)$ the critical point $x^* = 0.6658 \pm 0.0040$.

B. The index ν

We write Eq. (3) in a more convenient form:

$$G(x, N) = N^{-1} g \left[\left(\frac{N}{\xi_0} \right)^{1/\nu} (x^* - x) \right].$$

This implies

$$\left. \frac{\partial G}{\partial x} \right|_{x=x^*} = N^{-1+1/\nu} \times \text{const} \quad (5)$$

or

$$\nu = \left[1 + \ln \left(\frac{\partial G(x^*, N_1)}{\partial x^*} / \frac{\partial G(x^*, N_2)}{\partial x^*} \right) / \ln(N_1/N_2) \right]^{-1}.$$

The derivatives involved are readily calculable from the polynomial fits. We find, for $Z(2)$, $\nu = 0.935 \pm 0.001$ and for $Z(3)$, $\nu = 0.796 \pm 0.030$.

C. The susceptibility

The susceptibility of an infinite system is given by²

$$\chi(x) \sim - \left. \frac{\partial^2 F(x)}{\partial h^2} \right|_{h=0},$$

where F is the ground-state energy density and h is the magnetic field that enters into the Hamiltonian through the operator $\sum_i h\sigma_x(i)$. For Ising-type systems, χ is expected to diverge algebraically at x^* (Ref. 2):

$$\chi(x) \sim (x^* - x)^{-\nu} \text{ as } x \rightarrow x^*. \quad (6)$$

We define the finite system susceptibility $\chi(x, N)$ by

$$\chi(x, N) = - \left. \frac{\partial^2 F(x, N)}{\partial h^2} \right|_{h=0}.$$

The same scaling argument that we used for $G(x, N)$ can be repeated for $\chi(x, N)$ because Eqs. (2) and (6) have the same structure. Hence, we can use Eq. (3) for $\chi(x, N)$:

$$\chi(x, N) = N^{-p} f(N(x^* - x)^\nu / \xi_0), \quad (7)$$

where $f(y) \sim y^p$ as $y \rightarrow \infty$. p is determined so that Eq. (7) reduces to Eq. (6) in the limit $N \rightarrow \infty$, $x \rightarrow x^*$. One finds $p = -\gamma/\nu$. We rewrite Eq. (7) at $x = x^*$:

$$\chi(x^*, N) = \text{const} \times N^{\gamma/\nu},$$

or

$$\gamma = \nu \ln[\chi(x^*, N1)/\chi(x^*, N2)] / \ln(N1/N2).$$

Using this equation, we find for the $Z(2)$ model $\gamma/\nu = 1.735 \pm 0.011$. We have not computed γ for the $Z(3)$ model.

D. Possibility of using different variables

Instead of x and $G(x, N)$, derived from the dimensionless Hamiltonian W , we tried the customary variables⁴ $g = \sqrt{2/x}$ and the physical (dimensioned) mass gap $\Lambda(g, N) = (g/2a)G(x, N)$, derived from the physical (dimensioned) Hamiltonian $H = gW/2a$. In terms of these variables, Eq. (2) becomes

$$\Lambda(g) \sim \Lambda_0(g - g^*)^\nu, \text{ as } g \rightarrow g^*$$

where ν is the correlation length index: $\xi \sim c(g - g^*)^{-\nu}$. By using the same scaling arguments as before, but now in terms of g instead of x , we obtain

$$\Lambda(g, N) = N^{-1} f[(N/c)^{1/\nu}(g - g^*)]. \quad (8)$$

At $g = g^*$, Eq. (8) states that $\ln \Lambda(g^*, N)$ is linear in $\ln N$ with slope = -1, which is equivalent to saying that $\ln G(x^*, N)$ is linear in $\ln N$ with the same slope. Hence, the determination of x^* does not change. The value of ν , however, does change, because upon differentiating Eq. (8), we obtain at $g = g^*$,

$$\left. \frac{\partial \Lambda}{\partial g} \right|_{g=g^*} = N^{-1+1/\nu} \times \text{const} \quad (9)$$

or, in terms of G and x ,

$$G(x^*, N) - 2x^* \left. \frac{\partial G}{\partial x} \right|_{x=x^*} = N^{-1+1/\nu} \times \text{const},$$

which is different from Eq. (5). Using Eq. (9) one obtains, for $Z(2)$, $\nu = 0.996 \pm 0.004$ and for $Z(3)$, $\nu = 0.84 \pm 0.02$. These values of ν seem to be more in line with what one expects. It is not clear to us, though, why using g and Λ should work better than x and G .

IV. CONCLUSION

We have presented a method for solving the Hamiltonian formulation for lattice spin systems. Our work was inspired by the paper of Hamer and Barber,⁷ but we have modified their method by using the Lanczos scheme. The results of the scheme, when applied to $Z(2)$ and $Z(3)$, compare rather well to other methods, as can be seen in Tables I and II.

The method has two attractive features. The first feature is that it is easily generalizable to any model having a finite number of states per site. In fact only that part of the program that handles the spin flipping need be changed from one model to another. The rest of the program, which uses the Lanczos procedure, is essentially model independent. The method can also be applied to continuous spin systems if a truncation scheme is used to keep the number of states per site finite. An example of this would be the $Z(N)$ models, thought of as successive truncations of the $X - Y$ model.

The other feature is that *exact* results can be obtained by this method for finite systems. Finite systems may not be especially important for their own sake, but it is rather interesting that one can, using the Lanczos method, exactly solve a system like $Z(3)$ with 11 particles. In the most naive formulation, such a system would require dealing with $3^{11} = 177147$ states and diagonalizing a 177147

TABLE I. Comparison of our results for the fixed point (x^*), the correlation length index ν (from Sec. IIID), and the index γ/ν , for the $Z(2)$ model, with those obtained by an exact method (Ref. 11), perturbation-theory method (P. T.) (Ref. 2), and by a renormalization-group method (R. G.) (Ref. 5). The R. G. results are obtained by block-spinning 7 sites per cell. No direct computation of γ was done in Ref. 5.

Method	x^*	ν	γ/ν
Exact	1	1	1.750
P.T.	1	1	1.76
R.G.	1.053	0.86	
Our	0.9995	0.996	1.735

TABLE II. Comparison of our results for x^* and ν (from Sec. III D) for $Z(3)$, with those obtained (Ref. 12) experimentally (Expt.), by perturbation theory (P.T.) (Ref. 10), and by the renormalization group (Ref. 13), with two spins per block. The value of $x^* = \frac{2}{3}$, which we listed under P.T., is actually exact and can be obtained by a duality transformation [$Z(3)$ is self-dual].

Method	x^*	ν
Expt.		0.82
P.T.	$\frac{2}{3}$	0.84
R.G.	0.597	0.705
Our	0.666	0.84

$\times 177147$ matrix. With the Lanczos method, one encounters for the ground state of the same system states which are linear combinations of 5369 translationally invariant states and diagonalizes a 20×20 matrix.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under Grant No. NSF PHY 79-00272.

APPENDIX A

We considered the possibility of computing physical quantities for the infinite lattice directly rather than solving the finite one first and then scaling it.

We realize from the outset that exact results for the finite system will not serve this purpose because the finite system differs radically from the infinite one. It cannot, for example, have a phase transition even when the infinite one does. This implies that the finite-system mass gap $G(x, N)$ does not vanish at any finite x value, as depicted in Fig. 1, where we show $G(x, N)$ of $Z(2)$ for various N values as a function of x , along with the $N = \infty$ gap obtained exactly.¹¹ Figure 2 shows $G(x, N)$ of $Z(3)$ vs x .

The difference between a finite and an infinite system shows up computationally through the boundary conditions. The use of periodic boundary conditions at the edges allows information to propagate from one spin to one edge, loop around to the other edge, and return back to that spin. That is how a spin "knows" the system is finite. If we can somehow prevent this looping back of information, the spin would "think" the lattice is infinite and would behave accordingly. Since the potential energy flips two spins at a time, if we then stop the Lanczos process after $r = (N - 1)/2$ steps, V would have acted r times only and, thus, there

would not be enough interactions to flip a line of spins leading from one edge all the way to the other edge. This way these two spins would not be able to communicate. This is only a plausibility argument and one should not be surprised if this scheme of stopping the Lanczos process after r steps does not work.

In fact, we were surprised to find that for the $Z(2)$ case the values of the mass gap obtained in this way were identical to the exact results,¹¹ giving $\nu = 1$ exactly. We believe this is just an accident related to the fact that the high-temperature series for the mass gap terminates at second order,⁴ since, in fact, we also get exact results for the mass gap if we terminate the Lanczos process at any stage with fewer than $r = (N - 1)/2$ steps. Applied to $Z(3)$ the scheme does not yield good results.

APPENDIX B

In this Appendix we discuss some of the programming techniques that we used. The first problem one encounters is how to represent a quantum-mechanical state in the computer memory. For definiteness we will discuss $Z(2)$ -type states and generalize at the end. A typical translationally invariant $Z(2)$ state with $N = 3$ is

$$\frac{1}{\sqrt{3}} (|\downarrow\downarrow\uparrow\rangle + |\uparrow\uparrow\downarrow\rangle + |\uparrow\downarrow\downarrow\rangle).$$

We note that since all states in our calculation are translationally invariant, we need only store one of the three configurations $\downarrow\downarrow\uparrow$, $\uparrow\uparrow\downarrow$, and $\uparrow\downarrow\downarrow$ as a representative. One has to adopt some rule for picking out the representative. Our rule chooses $\downarrow\downarrow\uparrow$. The coefficient $1/\sqrt{3}$ need not be stored because we can always compute it from the representative. The computer memory is divided into cells (words) that can be addressed directly. Each word consists of a number of subcells called bits. A bit can store 1 or 0 only. A spin configuration like $\downarrow\uparrow\uparrow$ can be stored in the first 3 bits of a word as 011 if we take \uparrow to be represented by 1 and \downarrow to be represented by 0. The contents of the remaining bits are irrelevant. We see then that a translationally invariant $Z(2)$ state can be stored in one word by thinking of the first N bits of the word (counting from the right) as spins. Such a representation of translationally invariant states not only saves on memory, but is also very convenient as it makes it easy to act on the states with quantum operators. If, for example, we want to act on the state $\uparrow\uparrow\downarrow$, stored internally as 110 by the operator $\sigma_x(1)\sigma_x(2)$, which flips the first two spins (counting from the right), all we have to do is to form the logical **xor** (exclusive or) function

of the state with the number 3. This is because the number 3 is stored internally as 011 and $(011) \text{ XOR } (110) = (101)$, i.e.,

$$\sigma_x(1)\sigma_x(2)|\uparrow\uparrow\uparrow\rangle = |\uparrow\uparrow\uparrow\rangle.$$

In general, one can implement $\sigma_x(N)\sigma_x(N+1)$ by forming the XOR of the state with the number $3 \times 2^{N-1}$. Many other logical functions are available (AND, SHIFT, ...) so that any quantum operator can be implemented with a few instructions. These functions are standard (built in) at the assembly language level and, in most machines, are also available in high-level languages such as FORTRAN.

Having learned how to store simple translationally invariant states, we turn now to the most general $Z(2)$ state which is a linear combination of m states, each of which is translationally invariant. This state will be stored in two lists (arrays) of m words each: one to hold the spin configurations and the other to hold their coefficients. Consider, for example, the $N=3$ state

$$\frac{1}{(\alpha^2 + \beta^2)^{1/2}} \left(\alpha \frac{|\uparrow\uparrow\uparrow\rangle + |\downarrow\downarrow\downarrow\rangle + |\uparrow\downarrow\uparrow\rangle}{\sqrt{3}} + \beta \frac{|\downarrow\uparrow\uparrow\rangle + |\uparrow\downarrow\downarrow\rangle + |\uparrow\uparrow\downarrow\rangle}{\sqrt{3}} \right).$$

If we store the translationally invariant states under the word corresponding to its component with the smallest numerical value, the first list contains 001 in its first word and 011 in its second. The second list contains $\alpha/(\alpha^2 + \beta^2)^{1/2}$ in its first word and $\beta/(\alpha^2 + \beta^2)^{1/2}$ in its second. Note that in the Lanczos method only three states are present at any one time because to compute ψ_{n+1} from $H\psi_n$, we need ψ_{n-2} , ψ_{n-1} , and ψ_n .

The following points are helpful in writing the program:

(1) In order to compute ψ_{n+1} from $H\psi_n$, one has to perform scalar products and other operations that require searching the above-mentioned lists for a given record. The search time can be greatly reduced if the lists are ordered, because then one can use binary searching rather than search sequentially. We ordered the lists according to the numerical values of the words representing spin states.

(2) Another way to speed up the program is to write in assembly language the subroutines which manipulate the words representing quantum state vectors. We found that this would speed up the program by a factor of at least 2, and in some cases by a much larger factor.

(3) In the Lanczos method the coupling constant x enters the calculation in a very simple way so that one can perform most of the generation and manipulation of the lists without specifying the value of x . We were able to save a great deal of computer time by doing these manipulations only once and then storing the results on a disk. A different program can then read these results and use them to compute the eigenvalues at specified x values.

(4) Because of round-off errors, the state ψ_{n+1} may not be orthogonal to $\psi_1, \psi_2, \dots, \psi_n$. This can cause problems, as pointed out in Ref. 9. In that reference the authors overcame this problem by reorthogonalizing the states whenever a new state is generated. In all the calculations that we have done we did not encounter any such problem.

It is straightforward to generalize these ideas to any model with a finite number of states per site. For $Z(2)$ we reserved one bit for each site because the spin at that site can be either up or down. For $Z(n)$ one has to reserve l bits per site where $2^l \geq n$.

¹E. Fradkin and L. Susskind, Phys. Rev. D 17, 2637 (1978).

²J. B. Kogut, Rev. Mod. Phys. 51, 659 (1979).

³J. B. Kogut and L. Susskind, Phys. Rev. D 11, 395 (1975).

⁴C. J. Hamer, J. B. Kogut, and L. Susskind, Phys. Rev. D 19, 3091 (1979).

⁵S. Drell, M. Weinstein, and S. Yankielowicz, Phys. Rev. D 14, 487 (1976); R. Jullien, P. Pfeuty, J. N. Fields, and S. Doniach, Phys. Rev. B 18, 3568 (1978).

⁶J. E. Hirsch and G. F. Mazenko, Phys. Rev. B 19, 2656 (1979).

⁷C. J. Hamer and M. Barber, University of Melbourne and University of New South Wales report (unpub-

lished); see also, D. P. Crewther and C. Hamer, University of Melbourne report (unpublished).

⁸M. E. Fisher and M. N. Barber, Phys. Rev. Lett. 28, 1516 (1972).

⁹R. R. Whitehead, A. Watt, B. J. Cole, and I. Morrison, *Advances in Nuclear Physics* (Plenum, New York, 1977), Vol. 9, p. 123.

¹⁰S. Elitzur, R. B. Pearson, and J. Shigemitsu, Phys. Rev. D 19, 3698 (1979).

¹¹P. Pfeuty, Ann. Phys. (N. Y.) 57, 79 (1970).

¹²M. Bretz, Phys. Rev. Lett. 38, 501 (1977).

¹³H. H. Roomany and H. W. Wyld, calculations (unpublished).