Numerical Diagonalization of Quantum Spin Hamiltonians*)

Hidetoshi NISHIMORI and Yoshihiro TAGUCHI

Department of Physics, Tokyo Institute of Technology
Oh-okayama, Meguro-ku, Tokyo 152

(Received June 14, 1986)

We explain numerical methods to calculate low-lying eigenstates of the finite-size spin-1/2 anisotropic Heisenberg model. Sparse matrix techniques and a bit representation of spin configurations are shown to be quite useful in writing a program independent of specific lattice structure. A few examples of application are presented to demonstrate the effectiveness of our method.

§ 1. Introduction

One of the most versatile methods to investigate the localized spin models of magnetism is to numerically solve finite size systems on a computer. Numerical approaches include Monte Carlo simulations,[1] direct evaluation of the partition function[2)~4)] and numerical diagonalization of quantum spin Hamiltonians.[5),6)] For quantum spin systems the diagonalization method has long been the only possible approach, but recent developments in quantum Monte Carlo techniques[7)~9)] have aroused high activities in the investigation of finite-temperature properties of various quantum spin models.[7)~9)] However, the quantum Monte Carlo method still encounter difficulties at low temperatures,[9] and the diagonalization method plays a complementary role in this temperature region. We discuss the latter method in this paper. In the traditional numerical diagonalization of Hamiltonian, as represented by the pioneering work of Bonner and Fisher,[5] reduction of matrix dimension by using various symmetries of system has been essential because of the indispensable advantage of treating small size matrices on a computer of limited capacity (both memory size and speed). Each finite lattice has its own space (and other) symmetry; and therefore technical details to reduce matrix dimension have to be worked out for each system. For non-experts this fact has placed a non-negligible psychological barrier to the accessibility to this method. As a result the number of research groups working with this method is much less than that of groups employing Monte Carlo simulations.

Recent progress in computing facilities has greatly improved the situation. Both speed and memory capacity have reached the level which removes many of the above restrictions. For instance, the vector processor HITAC S-810/20 at the Computer Centre of the University of Tokyo has a main memory of 256 Mega bytes and executes 630 Mflops (million floating point operations per second) at its maximum speed. These numbers exceed those of conventional scalar machines by one to two orders of magnitude. Those features have enabled us to calculate lowest few eigenvalues and eigenstates of large-scale

*) Dedicated to Professor Takehiko Oguchi on the occasion of his retirement from Tokyo Institute of Technology.

matrices of dimension 10^5 to 10^6 at acceptable cost, without using space symmetry for matrix reduction if the number of spins N does not exceed 21 (the corresponding matrix dimension $\leq 352{,}716 = {}_{21}C_{10}$[6]). When we do not use space symmetry, the matrix is quite sparse (i.e., most of the elements are zero), and therefore we can make full use of well-developed computational techniques to treat sparse matrices including the Lanczos and conjugate gradient methods as explained in the following sections. Another important advantage of this approach is that one does not have to rewrite a computer program when he/she wishes to diagonalize a system of different lattice shape: Since no space symmetry is used to construct bases of a matrix, the lattice structure enters simply as data of connection between sites. In this way we were able to study various quantum spin systems at their ground and low excited states.[6),10)~13)] In this paper we review the basic ideas of our computational techniques as well as their representative applications to problems in quantum spin systems of current interest.

In the next section we explain the Lanczos method to calculate lowest few eigenvalues of a matrix. Section 3 treats the inverse iteration method supplemented by the conjugate gradient method for the evaluation of eigenstates. Section 4 is devoted to the explanation of practical techniques for efficient realization of prescriptions described in the preceding sections. A particular emphasis is put on a bit representation of the spin-1/2 system, in conjunction with matrix-vector multiplication without storing matrix elements beforehand. Several examples of application are presented in § 5 to demonstrate the effectiveness of the method. A brief summary is given in § 6.

§ 2. Eigenvalues

To diagonalize a matrix on a computer one usually tridiagonalize it first and then proceed to the final diagonalization. The Lanczos method[14)~16)] is known as an efficient algorithm to tridiagonalize a large sparse matrix. The procedure is as follows. If H is a matrix of interest, it should be transformed to a tridiagonal form T by a transformation matrix V : $T = V^{-1}HV$. Let us write the tridiagonal matrix T as

$$T = \begin{pmatrix} \alpha_1 & \beta_1 & & & & \\ \beta_1 & \alpha_2 & \beta_2 & & 0 & \\ & \beta_2 & \alpha_3 & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & 0 & & & \cdot & \beta_{n-1} \\ & & & & \beta_{n-1} & \alpha_n \end{pmatrix},$$

where n is the matrix dimension. Then the relation $T = V^{-1}HV$ or $VT = HV$ reads

$$\begin{aligned} H\boldsymbol{v}_1 &= \alpha_1\boldsymbol{v}_1 + \beta_1\boldsymbol{v}_2, \\ H\boldsymbol{v}_2 &= \beta_1\boldsymbol{v}_1 + \alpha_2\boldsymbol{v}_2 + \beta_2\boldsymbol{v}_3, \\ &\vdots \\ H\boldsymbol{v}_j &= \beta_{j-1}\boldsymbol{v}_{j-1} + \alpha_j\boldsymbol{v}_j + \beta_j\boldsymbol{v}_{j+1}, \\ &\vdots \\ H\boldsymbol{v}_n &= \beta_{n-1}\boldsymbol{v}_{n-1} + \alpha_n\boldsymbol{v}_n, \end{aligned} \tag{2·1}$$

where $(\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n)$ are the column vectors of V. The last relation of (2·1) can be interpreted as implying $\boldsymbol{v}_{n+1} = 0$ by comparison to the other relations in (2·1). The condition $\boldsymbol{v}_{n+1} = 0$ is satisfied if we construct $\boldsymbol{v}_j$ such that $\boldsymbol{v}_j$ is orthogonal to all previous $\boldsymbol{v}_i$'s $(i < j)$ because the $(n+1)$th orthogonal vector is not present in the n-dimensional space. This orthogonality can be achieved by choosing[15),16)]

$$\begin{aligned} \alpha_j &= {}^t\boldsymbol{v}_j H \boldsymbol{v}_j, \\ \beta_j &= \| H\boldsymbol{v}_j - \beta_{j-1}\boldsymbol{v}_{j-1} - \alpha_j\boldsymbol{v}_j \|. \end{aligned} \tag{2·2}$$

These equations (2·1) and (2·2) completely describe how to derive the tridiagonal form T from the given H. After tridiagonalization, the final diagonalization is performed by a standard routine such as the bisection method.[15),16)] (It should be remarked that a tridiagonal form occupies memory space only linear order of matrix dimension n and thus we do not have to invent special techniques to save memory when we proceed to the final diagonalization of the large-scale tridiagonal matrix.) As seen in (2·1), the main procedure in the Lanczos method is the matrix-vector multiplication. Since only the original matrix H appears in (2·1), we do not have to prepare large memory space for a transformed matrix as required in the Householder method.[15),16)] Thus a lot of memory space is saved.

In actual calculations of several lowest eigenvalues of a matrix, we do not have to obtain the full sequences of $\alpha_1, \cdots, \alpha_n$ and $\beta_1, \cdots, \beta_{n-1}$, but may terminate at α_j and β_{j-1} with $j \ll n$. In our experience only 50 to 100 iterations were necessary for convergence of the lowest two levels of a spin matrix up to 350,000 dimensions. The reason for this high efficiency lies in the iterative aspect of the Lanczos method (2·1). One generates a sequence of vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots$ by the relation (2·1), which defines the iteration process similar to the simple power method.[15)] The similarity is that a new vector $\boldsymbol{v}_{j+1}$ is generated from the preceding $\boldsymbol{v}_j$ by matrix multiplication. A distinct improvement over the power method is the subtraction of components parallel to the previous vectors from the newly generated vector (as implemented in the term $\beta_{j-1}\boldsymbol{v}_{j-1} + \alpha_j\boldsymbol{v}_j$ in (2·1)). That is, $\boldsymbol{v}_{j+1}$ is orthogonal to $\boldsymbol{v}_j, \boldsymbol{v}_{j-1}, \cdots, \boldsymbol{v}_1$. This aspect greatly accelerates convergence.

A weak point of the Lanczos process is its instability against rounding errors. The subtraction of components parallel to the previous vectors is not complete and $\boldsymbol{v}_i$ is not necessarily perfectly orthogonal to $\boldsymbol{v}_j$ $(j < i)$. This rounding error tends to mask eigenvalues of smaller absolute values since error-induced components of large eigenvalues grow much faster than those of small eigenvalues. However for the purpose of evaluating only several eigenvalues of largest absolute values (with negative sign in our case), this type of instability is not a difficulty. The only lesson is that one should not try to obtain all (or many) eigenvalues by the Lanczos method.

As the initial vector $\boldsymbol{v}_1$ we may choose an arbitrary vector as long as it has a non-vanishing component to the desired eigenstate (which is often the ground state). Otherwise the matrix multiplication does not yield the lowest eigenvalue. Since one does not know beforehand whether a vector has a non-vanishing component to the desired state, he/she has to try several different initial vectors to confirm convergence to the desired level.

§ 3. Eigenvectors

It is known[17] that the Lanczos method can be used to evaluate the eigenvector corresponding to the eigenvalue obtained in § 2. However it is sometimes necessary to calculate the eigenvector corresponding to an arbitrary eigenvalue (not necessarily the lowest level). We have thus developed a program to calculate the eigenvector corresponding to a given eigenvalue. The basic idea is the inverse iteration process explained in the following.

Suppose that $\boldsymbol{v}$ is the desired eigenvector corresponding to an eigenvalue e : $H\boldsymbol{v}=e\boldsymbol{v}$. If e_a is an approximate eigenvalue obtained by another method (such as the Lanczos method), the matrix $(H-e_aI)^{-1}$ (I denotes the unit matrix) has an eigenvalue $(e-e_a)^{-1}$ which is very large compared to other eigenvalues. Thus the sequence defined by

$$\boldsymbol{v}_{j+1}=(H-e_aI)^{-1}\boldsymbol{v}_j \tag{3·1}$$

rapidly converges to the eigenvector $\boldsymbol{v}$ unless the initial vector $\boldsymbol{v}_1$ is orthogonal to $\boldsymbol{v}$. The reason for fast convergence is that the successive multiplication of $(H-e_aI)^{-1}$ amplifies the component corresponding to $(e-e_a)^{-1}$ by orders of magnitude faster than those of other eigenvalues. In our experience less than five iterations of (3·1) were necessary for convergence with accuracy of several figures in coefficients of the eigenstate if we use e_a correct to several digits.

In implementing the process defined by (3·1), we have to solve

$$\bar{H}\boldsymbol{x}\equiv(H-e_aI)\boldsymbol{x}=\boldsymbol{b}, \tag{3·2}$$

where $\boldsymbol{b}$ denotes $\boldsymbol{v}_j$ (known vector) and $\boldsymbol{x}$ is for $\boldsymbol{v}_{j+1}$ (unknown). The solution of this equation can be found efficiently by the conjugate gradient (CG) method.[16),18] The original idea of the CG method is to minimize the error $f(\boldsymbol{x})={}^t\boldsymbol{s}\bar{H}\boldsymbol{s}$ ($\boldsymbol{s}=\bar{H}^{-1}\boldsymbol{b}-\boldsymbol{x}$, $\boldsymbol{x}$ here is an approximate solution of (3·2)) by varying $\boldsymbol{s}$: If $\boldsymbol{x}$ assumes the kth approximate value $\boldsymbol{x}_k$, the direction of steepest descent of $f(\boldsymbol{x})$ is parallel to $\boldsymbol{r}_k=\boldsymbol{b}-\bar{H}\boldsymbol{x}_k$ as can be verified by differentiating $f(\boldsymbol{x})$. (In this and the next paragraph $\boldsymbol{x}_k$ is the approximate solution of (3·2) which should not be confused with $\boldsymbol{v}_j$ in (3·1). The complete solution of (3·2) is still an approximate solution of the original eigenvalue problem.) By minimizing $f(\boldsymbol{x})$ along this line of steepest descent, we obtain the next approximation $\boldsymbol{x}_{k+1}$ as

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+({}^t\boldsymbol{r}_k\boldsymbol{r}_k)\boldsymbol{r}_k/{}^t\boldsymbol{r}_k\bar{H}\boldsymbol{r}_k. \tag{3·3}$$

Equation (3·3) defines the iteration process of the steepest descent method.[16),18] The CG method is an improvement of the steepest descent method in that the correction to the kth approximation $\boldsymbol{x}_k$ is chosen in the space orthogonal to all of the previous correction vectors $\boldsymbol{r}_{k-1}$, $\boldsymbol{r}_{k-2}$, ···. One starts from the kth set $\{\boldsymbol{x}_k, \boldsymbol{p}_k, \boldsymbol{r}_k\}$ and evaluate the $(k+1)$th approximation by

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+\alpha_k\boldsymbol{p}_k,$$

$$\boldsymbol{r}_{k+1}=\boldsymbol{r}_k-\alpha_k\bar{H}\boldsymbol{p}_k,$$

$$\boldsymbol{p}_{k+1}=\boldsymbol{r}_{k+1}+\beta_k\boldsymbol{p}_k,$$

where

$$\alpha_k={}^t\boldsymbol{p}_k\boldsymbol{r}_k/{}^t\boldsymbol{p}_k\bar{H}\boldsymbol{p}_k,$$

$$\beta_k=-{}^t\boldsymbol{r}_{k+1}\bar{H}\boldsymbol{p}_k/{}^t\boldsymbol{p}_k\bar{H}\boldsymbol{p}_k. \tag{3·4}$$

For proof of orthogonality of $\boldsymbol{p}_k$ to $\boldsymbol{p}_j$ ($j<k$) see Ref. 18). The important fact for our practical application is that the matrix $\bar{H}$ itself is not transformed but only vectors are successively changed. Thus the sparseness of matrix $\bar{H}$ is preserved in this process as it was in the Lanczos method.

In principle the iteration (3·4) terminates only after n steps (n is the matrix dimension) because no more than n orthogonal vectors can be generated in the n-dimensional space. In our practical applications, however, less than about 100 iterations gave sufficiently small correction ${}^t\boldsymbol{r}_k\boldsymbol{r}_k$ even for the largest matrices we have treated.

If the initial vector $\boldsymbol{v}_1$ is orthogonal to the correct eigenvector $\boldsymbol{v}$, it is impossible to reach $\boldsymbol{v}$ by multiplying $(H-e_aI)^{-1}$ to $\boldsymbol{v}_1$. Since we do not know beforehand whether $\boldsymbol{v}_1$ is orthogonal to $\boldsymbol{v}$ or not, it is necessary to try several different initial vectors as in the Lanczos method. This confirmation process serves also for counting the degree of degeneracy : If two orthogonal initial vectors converge to different eigenvectors with the same eigenvalue, then the level is at least doubly degenerate. Degeneracy of three and more eigenvectors can be checked in a similar manner.

§ 4. Practical techniques

In this section we present practical techniques in applying the methods of §§ 2 and 3 to the spin-1/2 matrices. The central idea is a matrix-vector multiplication without storing matrix elements using a bit representation of spin configurations. Let us first note that the Hamiltonian

$$H=2\sum J_{ij}\,(S_i^xS_j^x+S_i^yS_j^y+\Delta S_i^zS_j^z) \tag{4·1}$$

commutes with the z-component of the total spin $M_z=\sum_iS_i^z$. Hence H is block-diagonalized for each M_z and we may restrict ourselves to the space of a given M_z. This is the only symmetry we have used in our program. No space symmetry has been employed. In the space of a fixed M_z we adopt the following bit representation of a spin configuration : If the ith spin is up, then 1 is assigned to the ith bit of an integer m. A down spin at site i corresponds to 0 of the ith bit of m. Thus a spin configuration is represented by an integer m. Next we number allowed spin configurations in the space of the given M_z in increasing order of m. For instance, when $N=4$ and $M_z=0$, the configuration number $l=1$ is down-down-up-up ($m=3$ in decimal representation or 0011 in binary form) and number $l=2$ is down-up-down-up ($m=5$ or 0101) and so on. The maximum l in this example is 6 which is the number of combinations to choose 2 down spins out of 4 sites. Thus the matrix dimension n is 6. This correspondence between the configuration index l and the integer m is stored in an array (list 1). The inverse list (list 2) to find out the index l from a configuration m is stored in another array. The latter is needed when performing a matrix-vector multiplication.

After completion of indexing spin configurations we carry out the matrix-vector multiplication. The bit representation is quite suitable for doing it without explicitly

storing matrix elements beforehand. Suppose that we wish to operate the diagonal term $J_{ij}\Delta S_i^z S_j^z$ of the Hamiltonian (4·1) to a vector $\boldsymbol{v}$ (stored in an array). Let v_l denote the coefficient (amplitude) of the lth spin state m_l, where l runs from 1 to the matrix dimension n. We first have to see the ith and jth bits of m_l (the integer m_l is found out from list 1). If these bits have the same values, we add $\Delta J_{ij}v_l/4$ to the space saved for the lth coefficient of the new vector. Otherwise $-\Delta J_{ij}v_l/4$ should be added. This process is repeated for all pairs of sites (i, j) and for l from 1 to n. This completes the multiplication of the diagonal term of (4·1). The off-diagonal term $X_{ij}\equiv J_{ij}(S_i^+S_j^-+S_i^-S_j^+)$ is treated similarly. If the ith and jth bits of the lth integer m_l have the same values, the operator X_{ij} gives nothing and this step is skipped. If these bits are different, we exchange the concerned bits, i.e., (1, 0) to (0, 1) or (0, 1) to (1, 0), and find out the configuration index k of the resulting new integer m_k using the inverse list (list 2) as mentioned before. Then $J_{ij}v_k$ is added to the space for the lth coefficient of the new vector. This process is repeated for all pairs (i, j) and all configurations ($l=1$ to n). An apparent advantage of this method is that it saves the memory space for matrix elements and at the same time saves time for reading out matrix elements stored in an external memory device in conventional methods. A restriction is that only the spin-1/2 problem can be treated because of the full use of the bit representation. The necessary bit manipulations can be written in terms of FORTRAN bit functions, and all these functions are vectorized on HITAC S-810/20. Therefore the relative complication of bit manipulations (compared to the traditional stored-element multiplication) is not a significant drawback in the viewpoint of CPU time.

§ 5. Applications

We have applied the methods explained in the previous sections to various problems of quantum spin systems.[6),10)~13)] A review of some of our work is given here with emphasis on computational aspects.

The first example is the one-dimensional alternating antiferromagnet. It is known[19] that the spin degree of freedom of a group of organic compounds can be described by the Hamiltonian

$$H=J\sum_j\{1+(-1)^j\delta\}\boldsymbol{S}_j\cdot\boldsymbol{S}_{j+1},\quad (J>0) \tag{5·1}$$

where δ is called the alternation parameter. Let us denote the ground state energy of (5·1) by $E_0(N, \delta)$ and the first excited level by $E_1(N, \delta)$. The quantities of current interest[12] are the exponent of the energy gain

$$\lim_{N\to\infty}[E_0(N, \delta)-E_0(N, 0)]/N\sim\delta^a$$

and that of the energy gap

$$\lim_{N\to\infty}[E_1(N, \delta)-E_0(N, \delta)]\sim\delta^b.$$

Thus we have to evaluate the lowest two energy levels of the Hamiltonian (5·1). Since the isotropic Heisenberg model has all of its energy levels in the space of minimum M_z (the z-component of the total spin), we are allowed to diagonalize the matrix only in the

subspace of $M_z=0$ (note that N is even now). As the system size we have chosen $N=8$, 10, 12, ···, 20, and the alternation parameter δ assumes various values between 0 and 1. The diagonalization procedure explained in § 2 has been applied to each set of $N(\geq 10)$ and δ. In the case of $N=8$, the dimension of the space of $M_z=0$ is 70 which has turned out to be too small for the Lanczos tridiagonalization to work with sufficient stability. Thus, in this and all subsequent calculations, we have used the standard Householder tridiagonalization[15),16)] if the matrix dimensionality is less than 100. For larger matrices the iteration of the Lanczos method was continued until the second lowest (first excited) energy converges to the accuracy of 7 digits. The largest matrix has the dimensionality 184,756 ($={}_{20}C_{10}$). The CPU time for convergence of this matrix was about 30 seconds for one initial vector. Actually we have taken three orthogonal initial vectors to confirm that obtained energies are really the lowest two as explained in § 2. For smaller N the CPU time is proportional to the matrix dimension. The obtained energy levels have been analyzed by the finite-size scaling plots[12] to conclude $a=1.37\pm0.09$ and $b=0.79\pm0.06$.

The second example is on the antiferromagnetic system on the triangular lattice

$$H=2J\sum_{<ij>}(S_i^xS_j^x+S_i^yS_j^y+\Delta S_i^zS_j^z). \tag{5·2}$$

This system is predicted to have a quite unique ground state called the resonating valence bond (RVB) state.[20] We have thus tried to verify this prediction by inspecting the ground-state wave functions of (5·2) in finite-size samples. In the present case we have to obtain both the eigenvalue and eigenvector. As for the eigenvalue problem, statements in the preceding paragraph on the one-dimensional model apply almost without modifications. The only additional point is that the number of interacting pairs (bonds) is larger on the triangular lattice than in the one-dimensional case with the same system size N (same matrix dimension n). This difference is reflected to larger CPU time which is simply proportional to the number of bonds. The inverse iteration/conjugate gradient process was used to calculate the eigenvector once the lowest eigenvalue is found out by the Lanczos-bisection method. Again for small matrices ($n\leq100$) a more standard routine of LU decomposition[16] was employed at each stage of inverse iteration. The necessary CPU time was comparable to that of the Lanczos method. The results thus obtained show that the RVB state has a relatively large projection on the numerically exact wave function.[6] We have therefore verified that the RVB state describes fairly well the ground state of the model (5·2) as long as $N\leq21$.

We also examined field-dependent properties of an Ising-like antiferromagnet on the triangular lattice.[13] In the presence of an external magnetic field the ground state of the Hamiltonian

$$H=2J\sum_{<ij>}(S_i^xS_j^x+S_i^yS_j^y+\Delta S_i^zS_j^z)-h\sum S_i^z \tag{5·3}$$

does not necessarily belong to the space of the smallest M_z (i.e., 0 or 1/2). We have to calculate the lowest levels of the spaces of all possible M_z. Then at a given value of the field h the true ground state is chosen out of these energy levels. The value of M_z, in which space the true ground state is found, as a function of h gives the magnetization process of the system.[13] We have analyzed this magnetization process of the Ising-like antiferromagnet ($\Delta\geq1$, $J>0$) on the triangular lattice to check if the classical results[21] on this problem remain unchanged in the quantum mechanical system. When M_z differs

from 0, n (the dimensionality of matrix representation) is smaller than in the case $M_z=0$. We have found that the CPU time is simply proportional to n. From the obtained data we have verified quite classical character of ground state of the quantum system (5·3) for a range of external field $h_{c1}<h<h_{c2}$.

§ 6. Summary

We have explained numerical methods to calculate the eigenvalue and eigenstate of a spin-1/2 Heisenberg model of finite size lattice. The eigenvalue is calculated by the Lanczos method (and the bisection method). The eigenstate is obtained by the inverse iteration method. Each step of the inverse iteration process is carried out by the conjugate gradient method. In actual applications it is quite useful to apply a bit representation of spin configurations. Matrix-vector multiplication is then performed as matrix elements are generated, which saves much memory space (to be otherwise allocated to matrix elements). Accordingly all the necessary variables can be stored in the main memory even for the matrix of dimension up to 350,000, saving a lot of time for data exchange with external memory devices. Several instances of application have been explained with the intention to demonstrate effectiveness of our program. As mentioned in the Introduction, the present approach will continue to be a powerful tool in the investigation of the ground state and low excited states of quantum spin systems while quantum Monte Carlo simulations will keep to play a dominant role in the numerical study of finite-temperature properties.

Acknowledgements

This work is a by-product of innumerable inspiring discussions with Professor Takehiko Oguchi on the physics of quantum spin systems. We hereby express our appreciation and respect.

References

1) K. Binder and D. Stauffer, in *Applications of the Monte Carlo Methods in Statistical Physics*, ed. K. Binder, *Topics in Current Physics* 36 (Springer, Berlin, 1984), p. 1.
2) I. Morgenstern, in *Heidelberg Colloquium on Spin Glasses*, ed. J. L. van Hemmen and I. Morgenstern, *Lecture Notes in Physics* 192 (Springer, Berlin, 1983), p. 305.
3) M. Suzuki, Phys. Rev. **B31** (1985), 2975.
4) T. Oguchi and Y. Taguchi, Prog. Theor. Phys. Suppl. No. 87 (1986), 23.
5) J. C. Bonner and M. E. Fisher, Phys. Rev. **A135** (1964), 640.
6) T. Oguchi, H. Nishimori and Y. Taguchi, J. Phys. Soc. Jpn. **55** (1986), 323.
7) M. Suzuki, S. Miyashita and A. Kuroda, Prog. Theor. Phys. **58** (1977), 1377.
8) H. De Raedt and A. Lagendijk, Phys. Rep. **127** (1985), 233.
9) M. Takasu, S. Miyashita and M. Suzuki, Prog. Theor. Phys. **75** (1986), 1254, and references therein.
10) H. Nishimori, Y. Taguchi and T. Oguchi, J. Phys. Soc. Jpn. **55** (1986), 656.
11) T. Oguchi, H. Nishimori and Y. Taguchi, J. Phys. Soc. Jpn. **54** (1985), 4494.
12) K. Okamoto, H. Nishimori and Y. Taguchi, J. Phys. Soc. Jpn. **55** (1986), 1458.
13) H. Nishimori and S. Miyashita, J. Phys. Soc. Jpn. **55** (1986), No. 12.
14) R. R. Whitehead, A. Watt, B. J. Cole and I. Morrison, in *Advances in Nuclear Physics* Vol. 9, ed. M. Baranger and E. Vogt (Plenum, New York, 1977), p. 123.
15) J. H. Wilkinson, *The Algebraic Eigenvalue Problem* (Oxford, Oxford, 1965).

16) H. Togawa, *Matrix no Suchi Keisan (Numerical Calculation of Matrices)* (in Japanese) (Ohm-Sha, Tokyo, 1971).
17) E. R. Gagliano, E. Dagotto, A. Moreo and F. C. Alcaraz, Phys. Rev. **B34** (1986), 1677.
18) M. R. Hestenes, in *Proc. of Sixth Symposium in Applied Mathematics of AMS*, Santa Monica, 1953, ed. J. H. Curtiss (McGraw-Hill, New York, 1956), p. 83.
19) J. W. Bray, L. V. Interrante, I. S. Jacobs and J. C. Bonner, in *Extended Linear Chain Compounds* Vol. 3, ed. J. S. Miller (Plenum, New York, 1983), p. 353.
20) P. W. Anderson, Mat. Res. Bull. 8 (1973), 153.
21) S. Miyashita, J. Phys. Soc. Jpn. **55** (1986), 3605.