JOURNAL OF COMPUTATIONAL PHYSICS 114, 257–264 (1994)

The Method of Minimal Representations in 2D Ising Model Calculations*

BERESFORD N. PARLETT AND WEE-LIANG HENG

Department of Mathematics and the Computer Science Division of the EECS Department, University of California, Berkeley, California 94720

Received May 19, 1992

We present a new method for approximating the partition function of 2D Ising models using a transfer matrix of order 2^n. For $n = 30$ our current program took about 20 s on a Sparc station to obtain four correct decimals in the top two eigenvalues and 5 min for six correct decimals. Eigenvectors were computed at the same time. The temperature was within 3% of critical. The main idea is to force certain entries in vectors to have the same values and to find the crudest representation of this type that delivers the required accuracy. At no time does our program work with vectors with 2^n entries.

1. INTRODUCTION

The Ising model was proposed to explain properties of ferromagnets but since then it has found application to topics in chemistry and biology as well as in physics. For any reader unfamiliar with the model an excellent introduction targeted at a general audience is [Cip87]. The remainder of this section assumes some knowledge of the so-called transfer matrix. This paper presents a numerical method for computing properties of the 2D Ising model for given parameter values such as magnetic field strength B, temperature T, and coupling constants J. There are two avenues leading to such calculations: combinatorial and algebraic. Our method is in the second category which makes use of a transfer matrix M_n associated with a semi-infinite helical grid of "spins" or "sites" with n of them on each circular band. One form of M_n for $n = 3$ and $n = 4$, with the field strength B normalized with respect to the coupling constant J is as follows:

$$M_3 = \begin{bmatrix} a & & a^{-1} & & \\ b & & b^{-1} & & \\ & a & & a^{-1} & \\ & b & & b^{-1} & \\ & & b & & b^{-1} \\ & & c & & c^{-1} \\ & & & b & & b^{-1} \\ & & & c & & c^{-1} \end{bmatrix}$$

$$M_4 = \begin{bmatrix} a & & & a^{-1} & & & \\ b & & & b^{-1} & & & \\ & a & & & a^{-1} & & \\ & b & & & b^{-1} & & \\ & & a & & & a^{-1} & \\ & & b & & & b^{-1} & \\ & & & b & & & b^{-1} \\ & & & c & & & c^{-1} \\ & & & & b & & & b^{-1} \\ & & & & c & & & c^{-1} \\ & & & & & b & & & b^{-1} \\ & & & & & c & & & c^{-1} \\ & & & & & & b & & & b^{-1} \\ & & & & & & c & & & c^{-1} \end{bmatrix},$$

* The authors gratefully acknowledge partial support from Office of Naval Research Contact No. N00014-90-J-1372.

0021-9991/94 $6.00

where (with appropriate normalizations)

$$a = e^{(2-B)/T}, \qquad b = e^{-B/T}, \qquad c = e^{(-2-B)/T}.$$

The attractive property of M_n is that it is a nonnegative irreducible matrix whose dominant eigenvalue (called the Perron root) is the wanted partition function per spin. Thus it is only necessary to approximate this eigenvalue to the desired accuracy although the associated eigenvectors are also useful in approximating quantities of physical interest. Moreover, M_n is exceedingly sparse: it has exactly two nonzero entries per row (and column) arranged in a regular pattern. There is only one difficulty: M_n is of order 2^n and we are interested in the case $n - \infty$. We know of no calculations with $n \geqslant 20$ up until now.

Our approach uses a finite family $\{\mathscr{S}_{n,l}\}_{l=1}^{l=n}$ of orthogonal *indicial* vectors and approximates the top two column and row eigenvectors of M_n from the subspace spanned by $\mathscr{S}_{n,l}$.

Step 0. Initialize l to 1.

Step 1. Represent in compact form, the orthogonal projection P of the transfer matrix M_n onto the subspace $\text{span}(\mathscr{S}_{n,l})$.

Step 2. Compute the two largest eigenvalues and the associated column eigenvectors of P. These are, in a sense, the best approximations from the given indicial subspace $\text{span}(\mathscr{S}_{n,l})$. However they may not be good enough.

Step 3. Evaluate residual norms, condition numbers, and associated error bounds and estimates. If the estimates are satisfactory then compute the required properties of the model and stop. Otherwise return to Step 1 with the next member of each family, i.e., increase l by 1.

Our goal is to creep up to the coarsest of our vector representations that permits approximations of the desired accuracy. This minimal representation, which is not known in advance, gave us the name for our approach.

Note that the difficulty lies not in M_n itself but in the representation of vectors in $\mathbf{R}^{2^n}$. Indeed the special structure of M_n would permit evaluation of $M_n v$ for any 2^n-dimensional vector v with great efficiency. However a procedure that costs $O(2^n)$ may be too much when n is large and our central problem is the representation of vectors in $\mathbf{R}^{2^n}$.

Sparse vectors occur in sparse matrix work and Fuchs [Fuc89], when applying the *power method* to M_n, keeps only the largest 1000 entries of each vector. This device is satisfactory deep within the ferromagnetic region of the model. However, after studying the Perron vector in cases near the critical temperature we found that it contained almost no small entries. In different language, every configuration in the "spin" array contributes significantly to the partition function.

As a substitute for sparsity we propose to limit the number of distinct values that can occur among a vector's components. We do this by means of a family of "indicial vectors." Here is a sketch of the idea. More details are given in Section 2.

A vector in $\mathbf{R}^{2^n}$ may be thought of as a function on $\{1, 2, ..., 2^n\}$. What we call an indicial function is really a partition of this index set into disjoint subsets on each of which the vector is constant. Thus the vector takes on fewer than 2^n distinct values, perhaps only a few million of them. This sort of vector recalls Lebesgue's approach to integration via step functions. For a given partition f the set of all representable vectors forms a subspace $\mathscr{S}_f$ of $\mathbf{R}^{2^n}$. We bow to the influence of computer science and start counting at zero. If $\{e_0, ..., e_{2^n-1}\}$ denotes the standard basis and if $\{15, 93, 214, 866\}$ is one subset in the partition f then $e_{15} + e_{93} + e_{214} + e_{866}$ is one member of a natural orthogonal basis for $\mathscr{S}_f$. In other words, the natural basis vectors of $\mathbf{R}^{2^n}$ are aggregated according to f to produce an orthogonal basis of $\mathscr{S}_f$. An important feature of our approach is that these basis vectors are never represented explicitly in the computer. Careful index manipulation takes their place. Moreover, our choice of f yields a manageable representation of the projection P_f of M_n onto $\mathscr{S}_f$. P_f is nonnegative and irreducible. P_f is not as sparse as M_n but we hold it in a compact form that permits the efficient formation of $P_f w$ for appropriate w.

There is some freedom in the choice of the family of f's. Our f's are a compromise between physics and the very special structure of M_n. Details are given in Section 2.

The next task is to find the Perron vectors of P_f. Recall that the top two eigenvalues of M_n coalesce as the temperature becomes critical. We have used two approaches:

(a) a block power method with a block size of 2,

(b) a nonsymmetric Lanczos code.

The details are given in [PH92]. It turns out that it pays to compute the two largest eigenvalues together with their column and row eigenvectors. The reason that conventional techniques such as these are appropriate is that with our current indicial functions f, $\dim \mathscr{S}_f = O(n^2 2^{l-1})$ and so P_f is of modest order. In addition we form and compute similar quantities for $Q_{f'}$, the (orthogonal) projection of M_n^* onto an associated subspace $\mathscr{S}_{f'}$. The extra information from $Q_{f'}$ allows us to compute an approximate Perron row vector y^* to match the Perron column vector x for P_f. P_f and $Q_{f'}$ share the same Perron root. Fortunately $Q_{f'}$ is diagonally similar to P_f and need not be represented explicitly.

We would prefer to use the oblique projection of M_n onto the pair of subspaces $(\mathscr{S}_f, \mathscr{S}_{f'})$ but we have not yet found a convenient (sparse) representation because some of the canonical angles between $\mathscr{S}_f$ and $\mathscr{S}_{f'}$ equal $\pi/2$ and this fact complicates the representation.

Associated with the vectors x $(P_f x = x\pi)$ and y^*

($y^*Q_{f'} = \pi y^*$) are vectors $z_f \in \mathcal{S}_f$ and $w_f^* \in \mathcal{S}_{f'}$ that approximate the eigenvectors we seek. It is essential to be able to bound or estimate the accuracy of our approximate eigentriple (π_f, z_f, w_f^*).

Fortunately by using our special bases in $\mathcal{S}_f$ and $\mathcal{S}_{f'}$ appropriately we can compute (exactly in exact arithmetic) the associated residual vectors

$$r_f := M_n z_f - z_f \pi_f, \qquad s_{f'} := M_n^* w_{f'} - w_{f'} \pi_{f'}$$

and

$$\omega_f := w_f^* z_f / (\|w_f\|_2 \, \|z_f\|_2).$$

Although $r_f \in \mathbf{R}^{2^n}$, $s^{f'} \in \mathbf{R}^{2^n}$ we can accumulate $\|r_f\|^2$ and $\|s^{f'}\|^2$ and w_f during the computation of z_f and w_f^* and thus avoid ever having to store them. This is a key feature of the efficiency of our method. From $\|r\|$, $\|s\|$, and ω_f we can compute error bounds and error estimates. This is discussed in [PH92].

It is likely that our error estimates indicate that z_f, $w_{f'}$, and π_f are not sufficiently accurate. In that case we pick the next indicial function $\bar{f}$ in our family so that $\bar{f}$ is a refinement of f and $\mathcal{S}_f \subset \mathcal{S}_{\bar{f}}$, dim $\mathcal{S}_{\bar{f}} \approx 2$ dim $\mathcal{S}_f$. Then we repeat the cycle of approximations until the accuracy requirement is met or our resources are exhausted. This is not an iterative method because, in a finite number of steps, the indicial function becomes the identity.

By creeping up to adequate approximations from below we ensure that we end up with the coarsest indicial function that meets the given tolerance. In this way do we achieve the minimal representation, from our family, that gives our method its name. It is worth repeating that at no time in the cycle do we need to store a vector with 2^n components.

Quantities of interest are usually partial derivatives of the partition function. If we used differences to estimate derivatives that would sharply increase the required accuracy of our approximations. Fortunately Gartenhaus [Gar83] and Fuchs [Fuc89] have shown that some of the quantities of interest may be expressed in terms of z and w^* and so there is no need to use differences. This increases the scope of our approach significantly.

2. PROJECTIONS OF DUODIAGONAL MATRICES

Figures 3 and 4 show that the Perron eigenvector of M_8 may be approximated quite well by vectors in which certain positions are constrained to carry the same value. The challenge is, of course, to specify in general the right positions. Our approach makes heavy use of the binary representation of the numbers 0, 1, ..., $2^n - 1$. In particular we always use n-bits in a representation. Thus the positions in a 2^n-vector are indexed by the bit strings $\{00\cdots0.00\cdots01\cdots11\cdots11\}$. If we think of these strings as configurations of a 1D Ising model then we obtain equal contributions to the (total) partition function from all configurations (strings) which have the same k (the number of 1's) and t (the number of transitions). Thus we can group the bit strings by the values of (k, t). For $n = 4$ this partition is shown in Fig. 1.

# 1's	# transitions	indicial sets	indicial vectors
0	0	{0000}	ϵ_0
1	1	{1000.0001}	$\epsilon_8 + \epsilon_1$
1	2	{0100.0010}	$\epsilon_4 + \epsilon_2$
2	1	{1100.0011}	$\epsilon_{12} + \epsilon_3$
2	2	{1001.0110}	$\epsilon_9 + \epsilon_6$
2	3	{1010,0101}	$\epsilon_6 + \epsilon_5$
3	1	{1110.0111}	$\epsilon_{14} + \epsilon_7$
3	2	{1101.1011}	$\epsilon_{13} + \epsilon_{11}$
4	0	{1111}	ϵ_{15}

FIG. 1. Indicial sets and basis vectors for bit strings of length $n = 4$.

The approximation is not bad at the visual level but may well not be adequate. We need a systematic way to refine the partition induced by k and t. Our choice may not be optimal but it has the practical virtue of exploiting the duodiagonal structure of M_n. We use the last l bits in the string, for $l = 1, 2, ...$, as a refining parameter. For large enough l the original index set is recovered. The partition for $n = 5$, $l = 1$ is shown in Fig. 2. In each figure the last column lists a set of orthogonal vectors in $\mathbf{R}^{2^n}$ which we call

trailing bit	# 1's	# transitions	indicial sets	indicial vectors
0	0	0	{00000}	ϵ_0
0	1	1	{10000}	ϵ_{16}
0	1	2	{00010.00100.01000}	$\epsilon_2 + \epsilon_4 + \epsilon_8$
0	2	1	{11000}	ϵ_{24}
0	2	2	{00110.01100}	$\epsilon_6 + \epsilon_{12}$
0	2	3	{10010.10100}	$\epsilon_{18} + \epsilon_{20}$
0	2	4	{01010}	ϵ_{10}
0	3	1	{11100}	ϵ_{28}
0	3	2	{01110}	ϵ_{14}
0	3	3	{10110,11010}	$\epsilon_{22} + \epsilon_{26}$
0	4	1	{11110}	ϵ_{30}
1	1	1	{00001}	ϵ_1
1	2	1	{00011}	ϵ_3
1	2	2	{10001}	ϵ_{17}
1	2	3	{00101,01001}	$\epsilon_5 + \epsilon_9$
1	3	1	{00111}	ϵ_7
1	3	2	{10011,11001}	$\epsilon_{19} + \epsilon_{25}$
1	3	3	{01011,01101}	$\epsilon_{11} + \epsilon_{13}$
1	3	4	{10101}	ϵ_{21}
1	4	1	{01111}	ϵ_{15}
1	4	2	{10111,11011,11101}	$\epsilon_{23} + \epsilon_{27} + \epsilon_{29}$
1	5	0	{11111}	ϵ_{31}

FIG. 2. Indicial sets and basis vectors for $n = 5$ and $l = 1$.

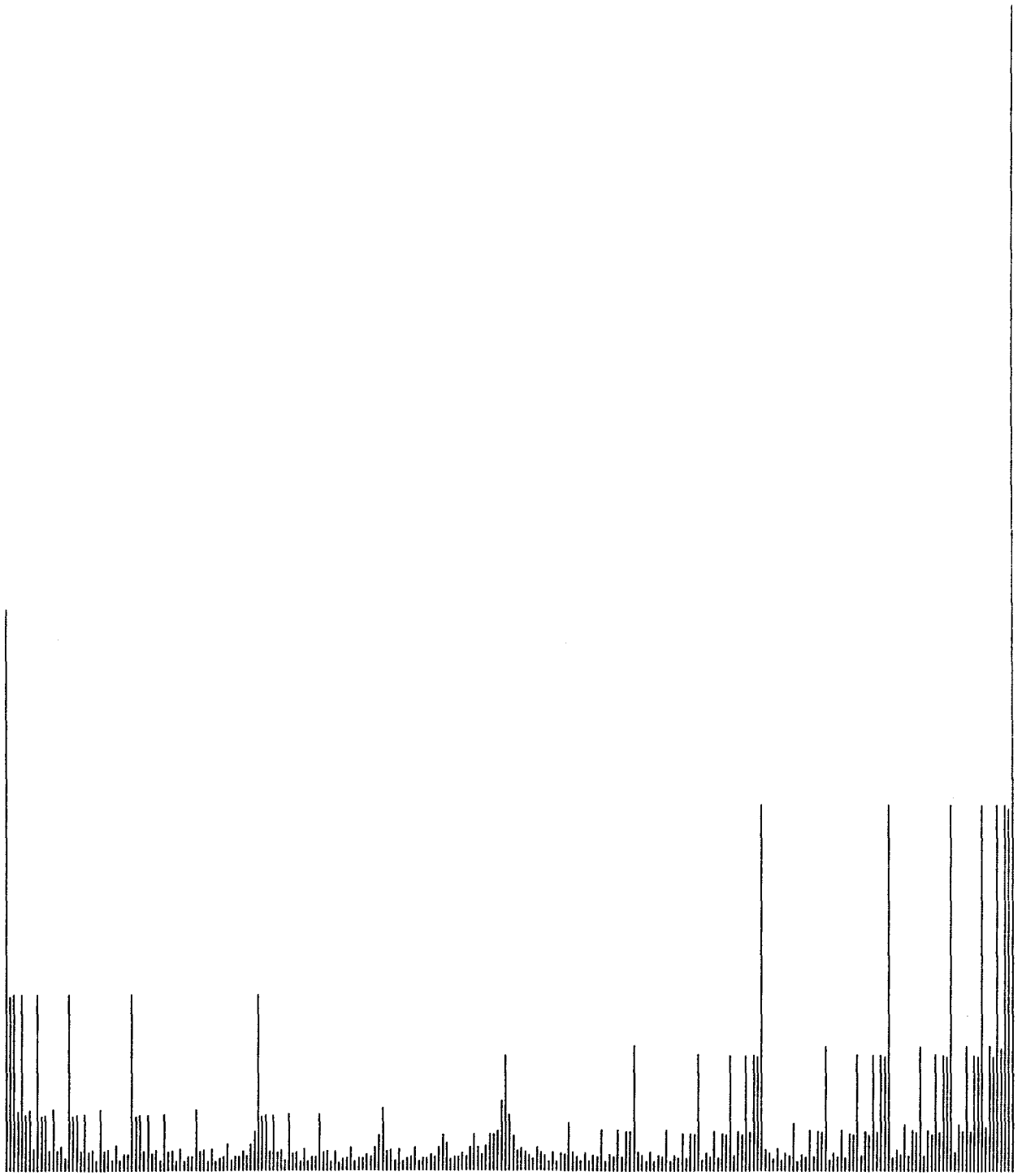

FIG. 3. Dominant column eigenvector of M_n, $n = 8$, $B = 0.0001$, $T = 1.6$.

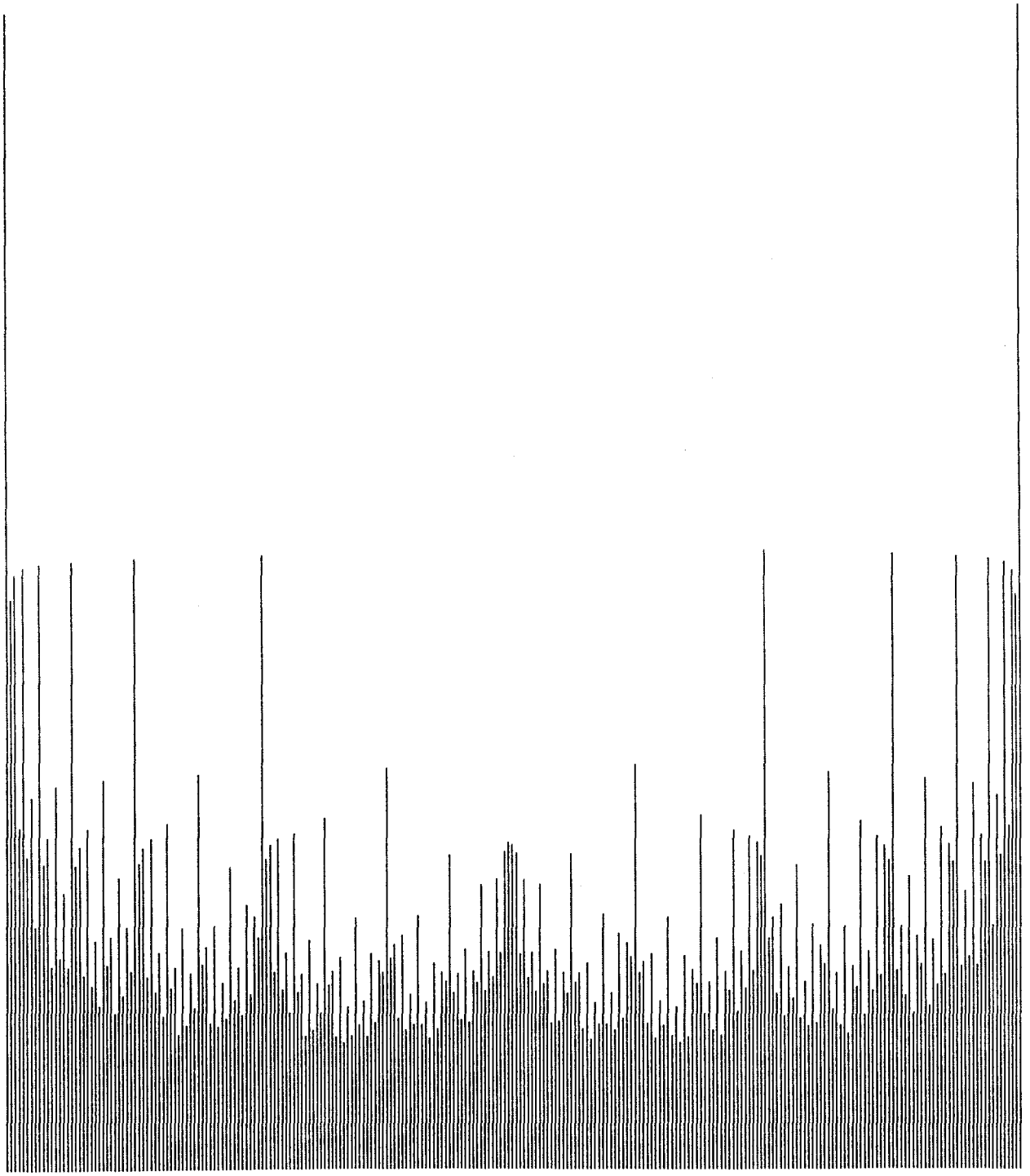

FIG. 4. Dominant column eigenvector of M_n, $n = 8$, $B = 0.0001$, $T = 2.2$.

$\mathscr{S}_{n,l}$: $n=4$, $l=0$ for Fig. 1; $n=5$, $l=1$ for Fig. 2. A careful analysis (see [Hen91]) shows the cardinality of this set,

$$|\mathscr{S}_{n,l}| = 2^l\left(1+\frac{(n-l+1)(n-l)}{2}\right) < n^2 2^{l-1}.$$

Our method's utility depends on obtaining adequate accuracy for small values of l. However, defining $\mathscr{S}_{n,l}$ is not enough. We need to derive the projection $P^C_{n,l}$ of M_n onto span($\mathscr{S}_{n,l}$) without actually using the indicial vectors exhibited in the figures above. Instead, by analysis, we determine a priori the positions and values of the nonzeros in $P^C_{n,l}$.

The duodiagonal form of M_n is essential to the analysis. The key fact is that the action of M_n on any vector in $\mathscr{S}_{n,l}$ can be expressed as a linear combination of either two or four vectors in $\mathscr{S}_{n,l+1}$. To illustrate the idea we show how to obtain a column of $P^C_{n,l}$ for $n=5$, $l=1$.

Let ω denote the last l bits of an n-string and let $x_{\omega,k,t}$ denote the indicial vector specified by ω, k, and t, e.g., {00010, 00100, 01000} is the indicial set with $k=1$, $t=2$, $\omega=0$ (with $n=5$, $l=1$). Thus $x_{0,1,2}=e_2+e_4+e_8$. The duodiagonal structure requires that the nonzeros in column j occur in rows $2j \bmod 2^n$ and $(2j+1) \bmod 2^n$:

$$\begin{aligned} M_n x_{0,1,2} &= M_n e_2 + M_n e_4 + M_n e_8 \\ &= (ae_4 + be_5) + (ae_8 + be_9) + (be_{16} + ce_{17}) \\ &= a(e_4+e_8) + b(e_5+e_9) + be_{16} + ce_{17} \\ &= ax_{00,1,2} + bx_{01,2,3} + bx_{00,1,1} + cx_{01,2,2}. \end{aligned}$$

See the matrices illustrated in the Introduction for the meaning of a, b, and c. Next we must determine the triples (ω', k', t') such that the inner product

$$\langle x_{\omega',k',t'}, M_n x_{\omega,k,t}\rangle$$

does not vanish. We return to our example above and read off the results because all indicial vectors are pairwise orthogonal:

$$\begin{aligned} \langle x_{00,1,2}, M_n x_{0,1,2}\rangle &= a\,\|x_{00,1,2}\|^2 = 2a \\ \langle x_{01,1,2}, M_n x_{0,1,2}\rangle &= b\,\|x_{01,1,2}\|^2 = 2b \\ \langle x_{00,1,2}, M_n x_{0,1,2}\rangle &= b\,\|x_{00,1,1}\|^2 = b \\ \langle x_{01,2,2}, M_n x_{0,1,2}\rangle &= c\,\|x_{01,2,2}\|^2 = c \end{aligned}$$

All other entries in column (0, 1, 2) vanish. Since the indicial vectors are not normalized a diagonal scaling is necessary. The result is that the four nonzeros in column (0, 1, 2) of $P^C_{5,1}$ are just a, b, b, c and they occur in the indicated rows. Thus P is easily stored, in compact form, in a rectangular array as discussed in the next section. Some other columns of P will have only two nonzeros. In thef general case the precise form of $P^C_{n,l}$ is not easy to determine. However, the analysis has been done and the details may be found in [Hen92]. The result is that P may be computed with arithmetic effort proportional to $|\mathscr{S}_{n,l}|$ and with storage proportional to $|\mathscr{S}_{n,l+1}|$. By means of this combinatorial analysis the largest two eigenvalues of M_n may be approximated by the largest two eigenvalues of $P^C_{n,l}$ and this can be tackled by more conventional techniques. Table I is a sample of the sizes of the sets $\mathscr{S}_{n,l}$ and the maximal number of ones in an indicial vector $x = x_{\omega,k,t}$.

TABLE I

Combinatorial properties of suffix-based indicial sets

n	2^n	$\lvert\mathscr{S}_{n,2}\rvert$	$\max_{\substack{\omega,k,t \\ \lvert\omega\rvert=2}} \lvert\mathbf{I}^n_{\omega,k,t}\rvert$	$\lvert\mathscr{S}_{n,4}\rvert$	$\max_{\substack{\omega,k,t \\ \lvert\omega\rvert=4}} \lvert\mathbf{I}^n_{\omega,k,t}\rvert$
10	1024	148	20	352	6
20	1.05×10^6	688	8820	2192	2520
30	1.07×10^9	1628	5.95×10^6	5632	1.59×10^6

3. IMPLEMENTATION ISSUES

We discuss the data structures used for indicial sets and indicial vectors $x_{\omega,k,t}$ as well as efficient algorithms for manipulating coefficient vectors.

3.1. *Representation of Vectors*

In order to turn the set $\mathscr{S}_{n,l}$ of indicial vectors $x_{\omega,k,t}$ into a basis a total ordering on the index triples (ω, k, t) with $|\omega| = l$ is needed. Recall that ω is the l-bit suffix of an n-string. Such a triple is *legitimate* if the corresponding index set is nonempty (or, equivalently, if $x_{\omega,k,t}$ is a vector in $\mathscr{S}_{n,l}$). The ordering on legitimate triples uses the number $v(\omega)$ of which ω is the binary representation,

$$(\omega, k, t) \prec_l (\omega', k', t')$$

if and only if,

(a) $v(\omega) < v(\omega')$, or

(b) $v(\omega) = v(\omega')$ and $k < k'$, or

(c) $v(\omega) = v(\omega')$, $k = k'$ and $t < t'$.

We use $\Phi(\omega, k, t)$ for the ranking of (ω, k, t) under this ordering, i.e., Φ maps (ω, k, t) into a nonnegative integer.

Let X denote the matrix whose columns are the vectors in $\mathscr{S}_{n,l}$ in the prescribed order. Then any vector g in span($\mathscr{S}_{n,l}$) may be written as

$$g = X\hat{g},$$

where $\hat{g}$ is the *suffix-based coefficient vector* of g. Note that $g \in \mathbf{R}^{2^n}$ while $\hat{g} \in \mathbf{R}^{|\mathscr{S}_{n,l}|}$.

It is tempting to use (ω, k, t) as an index so that any $\hat{g}$ could be held in a 3D array of real numbers. The trouble

TABLE II

Results for $n = 10$, $B = 0.0001$, $T = 1.6$ (True Eigenvalue = 3.5189840135)

l	Approximation	GRQ	Dim	Time (s)
2	3.5189867614 (2.7×10^{-6})	3.5189822267 (-1.8×10^{-6})	148	0.8
3	3.5189842995 (2.9×10^{-7})	3.5189837782 (-2.4×10^{-7})	232	1.3
4	3.5189839756 (-3.8×10^{-8})	3.5189839519 (-6.2×10^{-8})	352	2.0

TABLE III

Results for $n = 10$, $B = 0.0001$, $T = 2.2$ (True Eigenvalue = 2.5922922453)

l	Approximation	GRQ	Dim	Time (s)
2	2.5925207946 (2.3×10^{-4})	2.5922407533 (-1.5×10^{-5})	148	0.8
3	2.5923360346 (4.4×10^{-5})	2.5921803640 (-1.1×10^{-4})	232	1.5
4	2.5922660120 (-2.6×10^{-5})	2.5922266644 (-6.6×10^{-5})	352	2.4

TABLE IV

Results for $n = 20$, $B = 0.0001$, $T = 1.6$

l	Approximation	GRQ	Approximation-GRQ	Dim	Time (s)
2	3.5189802741	3.5189759878	4.3×10^{-6}	688	5.2
3	3.5189780552	3.5189731775	4.9×10^{-6}	1232	8.3
4	3.5189775223	3.5189777525	-2.3×10^{-7}	2192	17.0
5	3.5189776100	3.5189775601	5.0×10^{-8}	3872	35.7
6	3.5189776241	3.5189776145	9.6×10^{-9}	6784	71.7
7	3.5189776408	3.5189777184	-7.8×10^{-8}	11776	132.0

TABLE V

Results for $n = 20$, $B = 0.0001$, $T = 2.2$

l	Approximation	GRQ	Approximation-GRQ	Dim	Time (s)
2	2.5875164697	2.5873011057	2.2×10^{-4}	688	6.8
3	2.5873559732	2.5871852423	1.7×10^{-4}	1232	5.2
4	2.5872924943	2.5872247888	-6.8×10^{-5}	2192	11.2
5	2.5868850538	2.5869016769	-1.7×10^{-5}	3872	17.3
6	2.5872576018	2.5872809894	-2.3×10^{-5}	6784	80.9
7	2.5872475229	2.5872981335	-5.1×10^{-5}	11776	76.7

TABLE VI

Results for $n = 30$, $B = 0.0001$, $T = 1.6$

l	Approximation	GRQ	Approximation-GRQ	Dim	Time (s)
2	3.5189798036	3.5189277829	5.2×10^{-5}	1628	11.8
3	3.5187421095	3.5187271685	1.5×10^{-5}	3032	16.6
4	3.5189765962	3.5189734194	3.2×10^{-6}	5632	50.9
5	3.5189754869	3.5189630814	1.2×10^{-5}	10432	101.5
6	3.5189767326	3.5189765436	1.9×10^{-7}	19264	213.3
7	3.5189774542	3.5189775232	-6.9×10^{-8}	35456	472.3

TABLE VII

Results for $n = 30$, $B = 0.0001$, $T = 2.2$

l	Approximation	GRQ	Approximation-GRQ	Dim	Time (s)
2	2.5865877396	2.5864247904	1.6×10^{-4}	1628	21.1
3	2.5864495960	2.5863367635	1.1×10^{-4}	3032	17.5
4	2.5863989389	2.5863409514	5.8×10^{-5}	5632	38.3
5	2.5863738510	2.5863429058	3.1×10^{-5}	10432	64.3
6	2.5863633205	2.5863620747	1.2×10^{-6}	19264	139.1
7	2.5863635130	2.5863831549	-2.0×10^{-5}	35456	316.5

with this scheme is that there are holes (illegitimate triples) and so it is a bit wasteful of storage and, even worse, every access to the array must check whether the index is legitimate. Instead we simply map the legitimate triples onto $\{0, 1, ..., |\mathcal{S}_{n,l}| - 1\}$ using Φ.

The first task is to obtain Φ given (ω, k, t). The definition of $\prec_l$ shows that t plays a minor role and so we only need a 2D array, based on $v(\omega)$ and k, which we call *index*. The needed entries in the $2^l \times (n+1)$ array *index* may be computed initially. Then

$$\Phi(\omega, k, t) = \text{index}[v(\omega)][k] + t.$$

The next task is to represent the inverse mapping; given a value for Φ find ω, k, and t. This is easily accomplished with three 1D arrays of length $|\mathcal{S}_{n,l}|$, one for each of ω, k, and t. These are initialized before the calculations begin.

3.2. *Advantage of the Indexing Scheme*

When computing the dominant eigenvectors of a projection matrix the array $\hat{g}$ is treated as a conventional vector in $\mathbf{R}^{|\mathcal{S}_{n,l}|}$ but when we need to know an entry of $g = X\hat{g}$ in $\mathbf{R}^{2^n}$ we proceed as follows. To find $g(i)$ first obtain μ, the n-bit binary representation of i, together with its l-bit suffix ω (the last l bits of μ). Next compute k and t from μ. Finally loop up $\Phi(\omega, k, t)$ as described above. Then

$$g(i) = \hat{g}(\Phi(\omega, k, t)).$$

It is these simple bit manipulations and table look-ups that enable us to avoid the use of an array with 2^n entries.

3.3. *The Projection Matrix*

As mentioned earlier $P^C_{n,l}$ has at most four nonzeros per row and so it contains somewhat less than 4 $|\mathcal{S}_{n,l}|$ nonzeros. We represent P as a sequence of packed columns in a 1D array *col-proj*. In addition two index arrays *col* and *row* are needed; the ith nonzero is in position (row(i), col(i)) and has the value col-proj(i). The vector $P^C_{,l}v$ is formed by taking the linear combination of P's columns with coefficients given by v.

3.4. *Application of the Transfer Matrix to Approximate Eigenvectors*

Although the use of suffixes ω to refine our model has no justification from physics it does have the virtue that the action of M_n on vectors in span($\mathcal{S}_{n,l}$) can be computed exactly (modulo round off error). Consequently the norms of residual vectors may be computed without storing arrays of length 2^n. The reason, briefly, is that M_n maps span($\mathcal{S}_{n,l}$) into span($\mathcal{S}_{n,l+1}$). Details are given in [PH92].

3.5. *Extracting Information from the Projections*

Even though $|\mathcal{S}_{n,l}| \ll 2^n$ it is essential to use a fairly efficient method to compute the two dominant eigenvectors of $P^C_{n,l}$ and the two dominant row eigenvectors of $P^R_{n,l}$ (which is diagonally similar to $P^C_{n,l}$ and so it does not need separate treatment). Although we only want the dominant eigenvector we consider the calculation of two eigenvectors to be essential for efficiency when the temperature in the Ising model is near critical and there is less and less separation between the two largest eigenvalues.

We have tried two methods: (a) the block power method with block size 2 (called subspace iteration by numerical analysts and structural engineers); (b) the unsymmetric Lanczos algorithm.

A block Lanczos algorithm with block size 2 would probably be more efficient but we have not developed a code for that yet. Indeed the unsymmetric Lanczos code is not yet a standard method but in our experiments it becomes increasingly better than the block power method as n increases beyond 15. More details about the implementation and our error estimates are given in [PH92].

4. NUMERICAL RESULTS

Here are the results from a preliminary code using the nonsymmetric Lanczos algorithm. For the hardest case, $n = 30$ and temperature within 3% of critical, it took about 20 s on a Sparc station to obtain the partition function to three decimal digits and about 5 min to obtain five decimal digits. In Tables II–VII, GRQ is the generalized Rayleigh quotient $y^*M_n x/y^*x$. The temperature $T = 1.6$ is deep within the ferromagnetic region; $T = 2.2$ is within 3% of the critical temperature.

REFERENCES

[Cip87] B. A. Cipra, *Am. Math. Monthly* **94**, 937 (1987).
[Fuc89] N. H. Fuchs, *J. Comput. Phys.* **83** (1), 201 (1989).
[Gar83] S. Gartenhaus, *Phys. Rev. B* **27** (3), 1698 (1983).
[Hen92] W.-L. Heng, Center for Pure and Applied Math. Tech. Report No. 545, January 1992.
[PH92] B. N. Parlett and W.-L. Heng, Center for Pure and Applied Math. Tech. Report No. 550, May 1992.