報告

スーパーコンピュータＳＸ－４利用報告（追加）

本センターでは、昨年度、利用者の方々にＳＸ-４をいろいろな角度からモニターしていただき、得られた結果やノウハウ（失敗例や成功例を含めて）を本センターのすべての利用者が共有できるように提供していただくことを目的として、スーパーコンピュータ SX-４モニターの募集を行いました。

この報告書は、本センターニュース 98-5 月号、98-11 月号に掲載しましたが、新たに次の方の報告書が提出されましたので掲載します。

3 次元 Ising モデルの密度行列繰り込み群

— 数値計算的側面について —

神戸大学大学院自然科学研究科 構造科学・物質機能専攻 講師 西野友年
http://quattro.phys.sci.kobe-u.ac.jp/nishino.html
大阪大学大学院理学研究科物理学専攻 大学院博士後期課程 奥西巧一
http://glimmung.phys.sci.osaka-u.ac.jp/okunishi/okusjis.html

1 はじめに: 形而上の問題

N 次元格子模型の統計力学は「場の理論」を解析する上で大切な研究分野の一つです。(.... ということに興味の無い方は、後は読み飛ばして次の章に進んで下さい。) 特に 3 次元格子模型の分配関数を求める解析的手法は、例外的な場合を除いて皆無に等しく、数値計算 (= 力技) の助けは必須と言えるでしょう。この目的で一般的に用いられているのはモンテカルロ法ですが「ネコも杓子もモンテカルロ」では面白く無いので、ここでは趣向を変えて「数値繰り込み群」を用いてみることにしました。(繰り込み群 = 情報圧縮と考えて下さい。) まあ、どのような手法にも得手不得手がありますから、一つの研究対象に複数の計算手法を準備しておくのも、悪いことではありません。

さて、数値繰り込み群の分野において近年最も目覚ましい発展の一つと言えば、(計算物理学業界の) 誰もが White による「密度行列繰り込み群」(Density Matrix Renormalization Group) を取り上げるでしょう。[1]「密度行列繰り込み群」という名称は、とても長ったらしいので、ふつうは略称 DMRG で呼ばれています。(情報メディアの業界で Design Methodology Research Group, Distributed Multimedia Research Group, Digital Modulation Research Group 等の紛らわしい用語も用いられていますので、混同しないように御注意!!) DMRG は、もともと 1 次元量子系 (とい呼ばれる物理系) の為の数値計算手法なのですが、著者達はこの手法を 2 次元古典系に拡張しました。[2] ぶっちゃけて言うと、1 次元, 2 次元と来れば次は 3 次元という、とても単純な発想で 3 次元古典系、特に 3 次元 Ising 模型に手を出し (てしまっ) た訳です。

本稿では、上記のような訳の分からない物理的議論はさておいて、研究の為に用いた数値計算に関する工夫についてまとめます。(物理的な所については、付録の論文 [3] を参照して下さい。) ついでに、もうひと言。「3 次元 Ising 模型を研究して何の現世利益がある?」とは質問しないで下さい。何しろ、形而上の問題ですから。

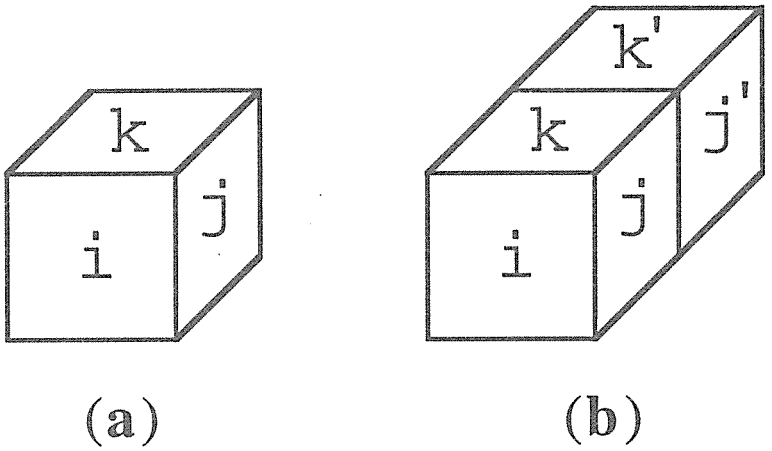

図 1: (a) 立方体への整数 i,j,k,l,m,n の張り付け。(b) 2 つの立方体の張り合わせ。

2 つづいて: 形而下の問題

まず、各面に (整数) 変数が張り付いている多面体を考えます。一番単純な例として、立方体 (= サイコロ) を考えましょう。立方体には X-, Y-, Z-面、及びそれぞれの裏面、合計 6 つの面がありますから、各面に整数 i,j,k,l,m,n を割り振れます。我々が注目するのは、ある整数の配列 $(ijklmn)$ が「出現する」確率 P_{ijklmn} です。ここで「出現する」と言ったのは統計力学的な意味においてなのですが、あまり深く考えずに、単に (正の) 実数値テンソル P_{ijklmn} を立方体に対応させて取り扱うとお考え下さい。仮にそれぞれの整数が 0 又は 1 のみを取るとした場合、$(ijklmn)$ は (000000) から (111111) までの 64 通りの違った組み合わせを取り得ます。従ってこの場合 P_{ijklmn} は (計算機上では) 64 次元ベクトル

$$\mathrm{P}((((((2*i+j)*2+k)*2+l)*2+m)*2+n) \tag{1}$$

として表現できます:「添え字は左側にある方が上位のアドレスを表す」という規則に基づいて 6 脚テンソルを一次元ベクトルに書き換えた訳です。(貴方はインテルが好きですか? それともモトローラが好きですか?)

さて、2 つの立方体 P_{ijklmn} と $P_{i'j'k'l'm'n'}$ の張り合わせを確率の合成則 (... というほど大層な物ではないのですが ^_^)

$$Q_{i(jj')(kk')l'(mm')(nn')} = \sum_{li'} P_{ijklmn} P_{i'j'k'l'm'n'} \tag{2}$$

で定義しましょう。左辺の Q は合成された直方体が、表面上に整数 $(i, 2j+j', 2k+k', l', 2m+m', 2n+n')$ を持つ確率です。面の対応は 図 1(b) に示してある通りで、

立方体の l に対応する面と i' に対応する面が「くっついて」消滅しています。(業界用語で「テンソルの縮約」と言います。) 式 (1) に習って Q をベクトルで表現す

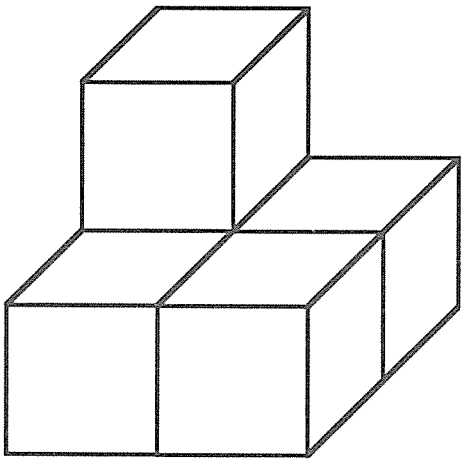

図 2: 積み木遊び。

ると

$$\mathtt{Q}((((((((i*2+j)*2+j')*2+k)*2+k')*2+l')*2+m)*2+m')*2+n)*2+n') \quad (3)$$

という 1024 次元ベクトルになります。 (.... こんな長ったらしい式は見たくもありませんが ^_^;) 同様に、どんどん立方体を積み重ねて行き、ルービックキューブのような物を作るのが「密度行列繰り込み群」の最終的な目標の一つです。従って、式 (2) のような計算を何回もくり返さなければなりません。(ちなみに、昨今の学生に「ルービックキューブ知ってるか?」と聞いても無駄です。NeXT Cube も同様。Generation Gap というモノでしょうか....)

さて、式 (2) をそのまま数値計算プログラムにすると、全部で 11 個の添え字がありますから、ループ長 2 の 11 重 do ループになります。このように短いループ長の多重 do ループは、 SX-4 (に限らずその辺に転がってるパイプライン型のスーパーコンピューター) にとって悪夢以外の何物でもありません。そこで、少し高速化 (ベクトル化) の工夫をしてみましょう。式 (2) のベクトル化が妨げられている原因は、Q の添え字の順番が $(ijkmnj'k'l'm'n')$ という素直な順番になっていなくて、Q を数値表現した場合のベクトル要素が

$$((((((((i*2+j)*2+k)*2+m)*2+n)*2+j')*2+k')*2+l')*2+m')*2+n') \quad (4)$$

という連続したアドレスに格納されていないからです。これを改善するべく、ちょっと高速化 (とプログラムの節約) の工夫してみましょう。

まず最初に P_{ijklmn} の足の順番を入れ替えたテンソルを準備します。

$$P'_{lmnijk} = P_{ijklmn} \quad (5)$$

これは、ただ単に P の要素を入れ替えただけです。この操作は、添え字を上位アドレス $ijk = (2*i+j)*2+k$ と、下位アドレス $lmn = (2*l+m)*2+n$ の 2 組に

分けて考えれば、2重 do ループで行えることが簡単にわかります。こうして得られた P'_{lmnijk} を用いて、

$$Q'_{mnijkj'k'l'm'n'} = \sum_{li'} P'_{lmnijk} P_{i'j'k'l'm'n'} \tag{6}$$

を先に計算します。添え字 $mnijk$ と $j'k'l'm'n'$ は連続した下位アドレスとみなせますので、式 (6) の計算はループ長 32 の 3 重 do ループとして計算できます。後は、$Q'_{mnijkj'k'l'm'n'}$ の添え字の順番を式 (5) のような代入により何回か入れ替えて、最終的に欲しい形 $Q_{i(jj')(kk')l'(mm')(nn')}$ に持って行くだけです。こういった添え字の入れ替えは高々 4 重 do ループで実行可能です。こうして我々は「魔の 11 重 do ループ」から逃れることができました。同時に、ベクトル化の恩恵により演算も高速化されます。

このように、添え字が幾つもあるテンソルの「縮約」や「内積」や「乗算」を行う場合は、予め要素の格納順番をベクトル化 (およびプログラミング) に都合の良いように変更しておいて、それから演算を開始するのが「ベクトル化による高速化」の秘訣です。(もうすぐ 21 世紀なのですが、こういう類いの高速化を自動的に行ってくれるコンパイラは、ついぞお目見えしませんでした。来世紀の発展に期待します。)

3 最後に: 残った課題

現在、計算速度のボトルネックになっているのは、式 (5) のようなベクトル格納形式の変更です。並列化で、より高速になるかな? と一瞬考えたのですが、結局は同じメモリーにアクセスに行ってしまうので、思ったほど並列化の効果は上がりません。もう少し精進して — 発想の転換をして — 更に数値計算を高速化できないかどうか、今後とも考えて行こうと思います。(さもないと「計算物理業界の若き獅子達」に一瞬で叩〜き潰されてしまいます。)

本研究の遂行に際して、SX モニタ制度による計算費免除をいただきました。この場をかりて大阪大学大型計算機センターに感謝いたします。また、付録の論文 (英文) は Journal of the Physical Society of Japan より許可を得て転載したものです。

参考文献

[1] S. R. White: Phys. Rev. Lett. 69 (1992) 2863; Phys. Rev. **B48**, 10345 (1993).

[2] T. Nishino: J. Phys. Soc. Jpn. **64**, 3598 (1995); 奥西巧一 (大阪大学 1999).

[3] T. Nishino and K. Okunishi, J. Phys. Soc. Jpn. **67**, 3066 (1998).

Journal of the Physical Society of Japan
Vol. 67, No. 9, September, 1998, pp. 3066-3072

A Density Matrix Algorithm for 3D Classical Models

Tomotoshi NISHINO* and Kouichi OKUNISHI[1,**]

Department of Physics, Faculty of Science, Kobe University, Rokkodai 657-8501
[1] *Department of Physics, Graduate School of Science, Osaka University, Toyonaka 560-0043*

(Received April 17, 1998)

We generalize the corner transfer matrix renormalization group, which consists of White's density matrix algorithm and Baxter's method of the corner transfer matrix, to three dimensional (3D) classical models. The renormalization group transformation is obtained through the diagonalization of density matrices for a cubic cluster. A trial application to 3D Ising model with $m = 2$ is shown as the simplest case.

KEYWORDS: density matrix, renormalization group, corner tensor

§1. Introduction

Variational estimation of the partition function has been one of the standard technic in statistical mechanics. For a two-dimensional (2D) classical lattice model defined by a transfer matrix T, the variational partition function per row is written as

$$\lambda = \frac{\langle V|T|V\rangle}{\langle V|V\rangle}, \tag{1.1}$$

where $|V\rangle$ represents the trial state and $\langle V|$ is its conjugate; λ is maximized when $\langle V|$ and $|V\rangle$ coincide with the left and the right eigenvectors of T, respectively. In 1941 Klamers and Wannier[1,2] investigated the Ising model, assuming that $|V\rangle$ is well approximated by a product of matrices

$$V(\ldots,i,j,k,l,\ldots) = \ldots F^{ij}F^{jk}F^{kl}\ldots, \tag{1.2}$$

where i, j, k, l, etc., are the Ising variables, and F^{ij} is a symmetric 2 by 2 matrix. The approximation is more accurate than both the mean-field and the Bethe approximations.[3] Baxter improved the trial state by introducing additional degrees of freedom.[4-6] His variational state is defined as

$$V(\ldots,i,j,k,l,\ldots) = \sum_{\ldots,a,b,c,d,\ldots} \ldots F^{ij}_{ab}F^{jk}_{bc}F^{kl}_{cd}\ldots, \tag{1.3}$$

where a, b, c, d, etc., are 2^n-state group spin variables. The tensor F^{ij}_{ab} contains $4\cdot 2^{2n}$ elements, and therefore it is not easy to optimize F^{ij}_{ab} — adjust the elements — so that λ is maximized. He solved the optimization problem by introducing the corner transfer matrix (CTM), and by solving self-consistent equations for the tensors.[6] In 1985 Nightingale and Blöte used Baxter's tensor product as a initial state in the projector Monte Carlo simulation for the Haldane system.[7] Baxter suggested an outline of generalizing his CTM method to 3D systems,[6] however, the project has not been completed.

Similar variational formulations have been applied to one-dimensional (1D) quantum systems, especially for $S = 1$ spin chains. The variational ground state $|\Psi\rangle$ is given by a modified tensor product

$$\Psi(\ldots,i,j,k,l,\ldots) = \sum_{\ldots,a,b,c,d,e,\ldots} \ldots A^i_{ab}A^j_{bc}A^k_{cd}A^l_{de}\ldots, \tag{1.4}$$

where the subscripts a, b, c, d, e, etc., are m-state group spin variables. Affleck, Lieb, Kennedy, and Tasaki (AKLT) showed that the ground-state of a bilinear-biquadratic $S = 1$ spin chain is exactly expressed by the tensor product with $m = 2$.[8] The variational formulation has been generalized by Fannes *et al.* for the arbitrary large m.[9-11] Now such ground state is called 'finitely correlated state'[10] or 'matrix product state'.[12] Quite recently Niggemann *et al.*[13] showed that the ground state of a 2D quantum systems can be exactly written in terms of a two-dimensional tensor product. Although $|\Psi\rangle$ in eq. (1.3) does not look like $|V\rangle$ in eq. (1.4), they are essentially the same. We can transform $|V\rangle$ into $|\Psi\rangle$ by obtaining A^i_{ab} from F^{ij}_{ab} through a kind of duality transformation;[14] the opposite is also possible.

The application of both $|V\rangle$ in eq. (1.3) and $|\Psi\rangle$ in eq. (1.4) are limited to translationally invariant (or homogeneous) systems. In 1992 White established a more flexible numerical variational method from the view point of the real-space renormalization group (RG).[15,16] Since his numerical algorithm is written in terms of the density matrix, the algorithm is called 'density matrix renormalization group' (DMRG). White's variational state is written in a position dependent tensor product[17,18]

$$\Phi(\ldots,i,j,k,l,\ldots) = \sum_{\ldots,a,b,c,d,e,\ldots} \ldots A^i_{ab}B^j_{bc}C^k_{cd}D^l_{de}\ldots, \tag{1.5}$$

where A^i_{ab} is not always equal to B^i_{ab}, etc. This inhomogeneous property in $|\Phi\rangle$ makes DMRG possible to treat open boundary systems[19] and random systems.[20] Now the DMRG is widely used for both quantum[21] and classical[22-24] problems. Quite recently Dukelsky

* E-mail: nishino@phys560.phys.kobe-u.ac.jp
** E-mail: okunishi@godzilla.phys.sci.osaka-u.ac.jp

et al. analyzed the correspondence (and a small discrepancy) between DMRG and the variational formula in eq. (1.4).[25-27]

The decomposition of the trial state into the tensor product tells us how to treat lattice models when we try to obtain the partition function. The essential point is to break-up the system into smaller pieces — like the local tensor F^{ij}_{ab} in eq. (1.3) or A^{i}_{ab} in eq. (1.4) — and reconstruct the original system by multiplying them. According to this idea, the authors combine DMRG and Baxter's method of CTM, and proposed the corner transfer matrix renromalization group (CTMRG) method.[28, 29] It has been shown that CTMRG is efficient for determinations of critical indices[29] or latent heats.[30]

The purpose of this paper is to generalize the algorithm of CTMRG to three-dimensional (3D) classical systems. We focus on the RG algorithm rather than its practical use. In the next section, we briefly review the outline of CTMRG. The key point is that the RG transformation is obtained through the diagonalization of the density matrix. In §3 we define the density matrix for a 3D vertex model, and in §4 we explain the way to obtain the RG transformation. The numerical algorithm is shown in §5. A trial application with $m = 2$ is shown for the 3D Ising Model. Conclusions are summarized in §6.

§2. Formulation in Two Dimension

The aim of CTMRG is to obtain variational partition functions of 2D classical models. Let us consider a square cluster of a 16-vertex model (Fig. 1) as an example of 2D systems. We impose the fixed boundary condition, where the boundary spins shown by the cross marks point the same direction. In order to simplify the following discussion, we assign a symmetric Boltzmann weight $W_{ijkl} = W_{jkli} = W_{ilkj}$ to each vertex,[31] where i, j, k and l denote two-state spins (= Ising spins or arrows) on the bonds. (Fig. 2(a))

We employ two kinds of transfer matrices in order to express the partition function Z_{2N} of the square cluster with linear dimension $2N$. One is the half-row transfer matrix (HRTM). Figure 2(b) shows the HRTM P^{i}_{ab} with length $N = 3$, where the subscripts $a = (a_1, a_2, \ldots, a_N)$ and $b = (b_1, b_2, \ldots, b_N)$, respectively, represent row spins — in-line spins — on the left and the right sides of the HRTM. We think of P^{i}_{ab} as a matrix labeled by the superscript i. The other is the Baxter's corner transfer matrix (CTM),[4-6] that represents Boltzmann weight of a quadrant of the square. Figure 2(c) shows the CTM C_{ab} with linear dimension $N = 3$. The partition function Z_{2N} is then expressed as

$$Z_{2N} = \mathrm{Tr}\,\rho = \mathrm{Tr}\,C^4, \tag{2.1}$$

where $\rho_{ab} \equiv \left(C^4\right)_{ab}$ is the density matrix. From the symmetry of the vertex weight W_{ijkl}, the matrices P^{i}_{ab}, C_{ab}, and ρ_{ab} are invariant under the permutation of subscripts.

There are recursive relations between W, P, and C. We can increase the length of HRTM by joining a vertex

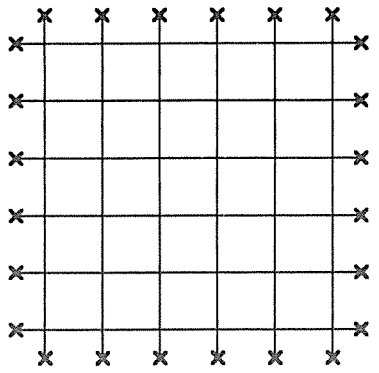

Fig. 1. Square cluster of a symmetric vertex model; the shown system is the example with linear dimension $2N = 6$. The cross marks × show the boundary spins.

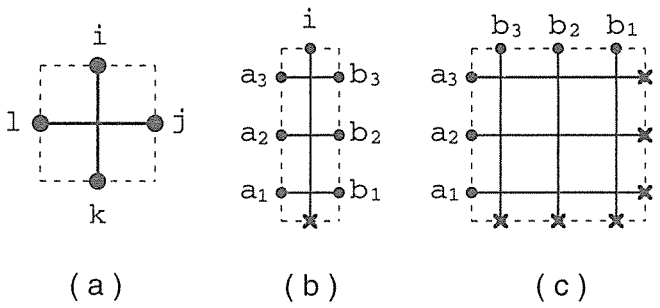

Fig. 2. Boltzmann weight and transfer matrices. The dots represent spin variables inside the square cluster shown in Fig. 1, and the cross marks represent the boundary spins. (a) Vertex weight W_{ijkl}. (b) Half-row transfer matrix P^{i}_{ab}. (c) Corner transfer matrix C_{ab}.

$$P^{\,i}_{\bar{a}\bar{b}} = \sum_{k} W_{ijkl} P^{\,k}_{ab}, \tag{2.2}$$

where the extended row-spins are defined as $\bar{a} = (a, l) = (a_1, a_2, \ldots, a_N, l)$ and $\bar{b} = (b, j) = (b_1, b_2, \ldots, b_N, j)$. (Fig. 3(a)) In the same manner, the area of CTM can be extended by joining two HRTMs and a vertex to the CTM

$$C_{\bar{a}\bar{b}} = \sum_{cd\ kj} W_{ijkl} P^{\,j}_{db} P^{\,k}_{ac} C_{cd}, \tag{2.3}$$

where the extended row-spins $\bar{a}$ and $\bar{b}$ are defined as $\bar{a} = (a, l) = (a_1, a_2, \ldots, a_N, l)$ and $\bar{b} = (b, i) = (b_1, b_2, \ldots, a_N, i)$. (Fig. 3(b)) In this way, we can construct HRTM and CTM with arbitrary size N by repeating the extension eqs. (2.2) and (2.3).

It should be noted that the matrix dimension of both C_{ab} and P^{i}_{ab} increases very rapidly with their linear size N. The fact prevents us to store the matrix elements of C_{ab} and P^{i}_{ab} when we numerically calculate the partition function Z_{2N}. This difficulty can be overcomed by compressing CTM and HRTM into smaller matrices via the density matrix algorithm,[15, 16] where the RG transformation is obtained through the diagonalization of the density matrix $\rho_{ab} \equiv (C^4)_{ab}$. Let us consider the eigenvalue equation for the density matrix

$$\sum_{b} \rho_{ab} \Lambda^{\alpha}_{b} = \lambda_{\alpha} \Lambda^{\alpha}_{a}, \tag{2.4}$$

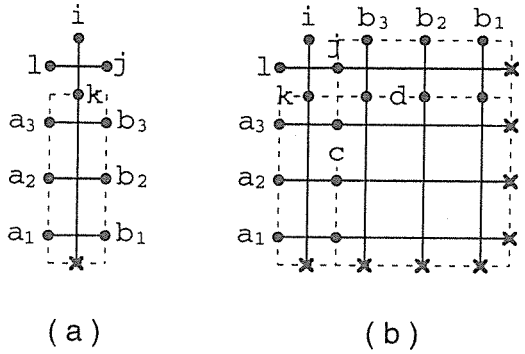

Fig. 3. Extensions of (a) the HRTM (eq. (2.2)), and (b) the CTM. (eq. (2.3)).

where λ_α is the eigenvalue in decreasing order $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$, and $\mathbf{A}^\alpha = (A_1^\alpha, A_2^\alpha, \ldots)^T$ is the corresponding eigenvector that satisfies the orthogonal relation

$$(\mathbf{A}^\alpha, \mathbf{A}^\beta) = \sum_a A_a^\alpha A_a^\beta = \delta^{\alpha\beta}. \tag{2.5}$$

It has been known that λ_α rapidly approaches to zero with respect to α,[6,16] and that we can neglect tiny eigenvalues from the view point of numerical calculation. We consider only m numbers of dominant eigenvalues in the following; the greek indices run from 1 to m. The number m is determined so that $\sum_{\alpha=1}^m \lambda_\alpha$ becomes a good lower bound for the partition function $Z_{2N} = \mathrm{Tr}\,\rho$.

Equation (2.4) shows that for a sufficiently large m the density matrix ρ can be well approximated as

$$\rho_{ab} \sim \sum_{\alpha=1}^m A_a^\alpha A_b^\alpha \lambda_\alpha. \tag{2.6}$$

The above approximation shows that the m-dimensional diagonal matrix

$$\tilde{\rho}_{\alpha\beta} = \sum_{ab} A_a^\alpha A_b^\beta \rho_{ab} = \delta_{\alpha\beta}\lambda_\beta \tag{2.7}$$

contains the relevant information of ρ; we can regard $\tilde{\rho}$ as the renormalized density matrix. This is the heart of the density matrix algorithm: *the RG transformation is defined by the matrix* $A = (\mathbf{A}^1, \mathbf{A}^2, \ldots, \mathbf{A}^m)$ *which is obtained through the diagonalization of the density matrix.*

As we have applied the RG transformation to the density matrix ρ, we can renormalize the CTM by applying the matrix A as

$$\tilde{C}_{\alpha\beta} = \sum_{ab} A_a^\alpha A_b^\beta C_{ab}. \tag{2.8}$$

Since C_{ab} and ρ_{ab} have the common eigenvectors - remember that $\rho = C^4$ - the renormalized CTM is an m-dimensional diagonal matrix

$$\tilde{C} = \mathrm{diag}(\omega_1, \omega_2, \ldots, \omega_m), \tag{2.9}$$

where ω_α is the eigenvalue of the CTM that satisfies $\lambda_\alpha = \omega_\alpha^4$. In the same manner, we obtain the renormalized HRTM

$$\tilde{P}^{\,i}_{\alpha\beta} = \sum_{ab} A_a^\alpha A_b^\beta P^{\,i}_{ab}. \tag{2.10}$$

In this case $\tilde{P}^{\,i}_{\alpha\beta}$ is not diagonal with respect to α and β; *the RG transformation is not always diagonalization.*

We can extend the linear size of CTM and HRTM using eqs. (2.2) and (2.3), and we can reduce their matrix dimension by the RG transformation in eqs. (2.7) and (2.8). By repeating the extension and the renormalization, we can obtain the renormalized density matrix $\tilde{\rho}$ and the approximate partition function $\tilde{Z}_{2N} = \mathrm{Tr}\,\tilde{\rho}$ for arbitrary system size N. This is the outline of the CTMRG.

§3. Density Matrix in Three Dimension

In order to generalize the density matrix algorithm to 3D systems, we first construct the density matrix in three dimension. As an example of 3D systems, we consider a 64-vertex model, that is defined by a Boltzmann weight W_{ijklmn}. (Fig. 4(a)) In order to simplify the following discussion, we consider the case where W_{ijklmn} is invariant under the permutations of the two-state spins i, j, k, l, m and n.[32] As we have considered a square cluster in two-dimension, (Fig. 1) we consider a cube with linear dimension $2N$, where the boundary spins (on the surface of the cube) are fixed to the same direction. According to the variational formulation shown in §1, we first decompose the cube into several parts shown in Figs. 4(b)–4(d).

The tensor $P^{\,i}_{abcd}$ shown in Fig. 4(b) is a kind of three-dimensional HRTM. The superscript i represents the two-state spin at the top. The spin at the bottom is fixed, because it is at the boundary of the system. The subscript a represents the group of in-line spins $a = (a_1, a_2, \ldots, a_N)$; b, c, and d are defined in the same way. From the symmetry of the vertex weight, $P^{\,i}_{abcd}$ is invariant under the permutations of subscripts.

The tensor S^{XY}_{ab} shown in Fig. 4(c) does not have its 2D analogue; it is an array of vertices. The subscripts a and b represent in-line spins; other two sides are the boundary of the cube. The superscript X represents an N by N array of spins on the square surface

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ x_{N1} & \cdots & \cdots & x_{NN} \end{pmatrix}, \tag{3.1}$$

where x_{NN} is closest to the center of the cube, and Y is the spin array on the other surface; x_{ij} and y_{ij} are connected to the same vertex at the position $\{ij\}$. The tensor is invariant under the permutation of X and Y ($S^{XY}_{ab} = S^{YX}_{ab}$), but is not invariant for a and b ($S^{XY}_{ab} \neq S^{XY}_{ba}$); S^{XY}_{ab} is equal to S^{ZW}_{ba} where $Z = X^T$ and $W = Y^T$.

Figure 4(d) shows the corner tensor C^{XYZ}, which is a kind of three-dimensional CTM.[6] The superscripts are defined in the same way as eq. (3.1). (The boundary spins on the surfaces of the original cube are fixed.) It should be noted that C^{XYZ} is not equal to C^{WYZ} where $W = X^T$, because each surface has its own orientation.

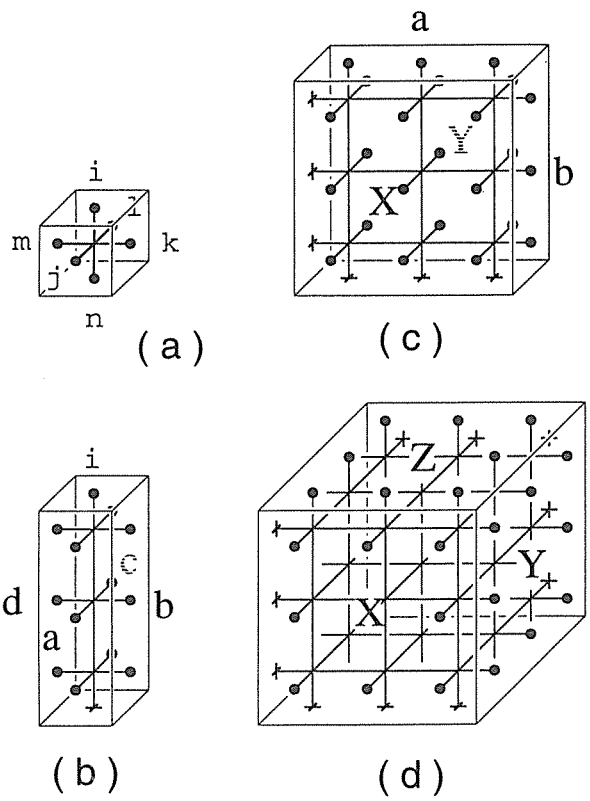

Fig. 4. Parts of the cubic cluster with linear dimension $2N$: (a) Vertex weight W_{ijklmn}. (b) The tensor P^{i}_{abcd}. (c) The tensor S^{XY}_{ab}. (d) Corner Tensor C^{XYZ}. The cross marks × represent the boundary spins.

Following the formulation in two-dimension, let us consider the size extension of P, S, and C. The length of P can be increased by joining a vertex (Fig. 5(a))

$$P^{\bar{i}}_{\bar{a}\bar{b}\bar{c}\bar{d}} = \sum_{n} W_{ijklmn}\, P^{n}_{abcd}, \tag{3.2}$$

where the extended in-line spins are defined as $\bar{a} = (a,j) = (a_1, a_2, \ldots, a_N, j)$, $\bar{b} = (b,k) = (b_1, b_2, \ldots, b_N, k)$, $\bar{c} = (c,l) = (c_1, c_2, \ldots, c_N, l)$, and $\bar{d} = (d,m) = (d_1, d_2, \ldots, d_N, m)$. The linear size of S can be increased by joining two P and a vertex (Fig. 5(b))

$$S^{\bar{X}\bar{Y}}_{\bar{a}\bar{b}} = \sum_{ln}\sum_{ce} W_{ijklmn}\, P^{n}_{abcd}\, P^{l}_{efgh}\, S^{XY}_{ce} \tag{3.3}$$

where the extended in-line spins are defined as $\bar{a} = (a,j) = (a_1, a_2, \ldots, a_N, j)$, and $\bar{b} = (g,i) = (g_1, g_2, \ldots, g_N, i)$. The extended spin array $\bar{X}$ is defined as

$$\bar{X} = \left(\begin{array}{ccc|c} x_{11} & \cdots & x_{1N} & f_1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{N1} & \cdots & x_{NN} & f_N \\ \hline b_1 & \cdots & b_N & k \end{array}\right), \tag{3.4}$$

and $\bar{Y}$ is defined in the same way from the indices m, d, h and Y. The linear size of the corner tensor C can be increased by joining three P, three S, and a vertex (Fig. 5(c))

$$C^{\bar{X}\bar{Y}\bar{Z}} = \sum_{lmn}\sum_{cd}\sum_{eh}\sum_{qr}\sum_{TUV} W_{ijklmn}\, P^{n}_{abcd}\, P^{l}_{efgh}\, P^{m}_{opqr} \times S^{XT}_{qd}\, S^{YU}_{ce}\, S^{ZV}_{hr}\, C^{TUV}. \tag{3.5}$$

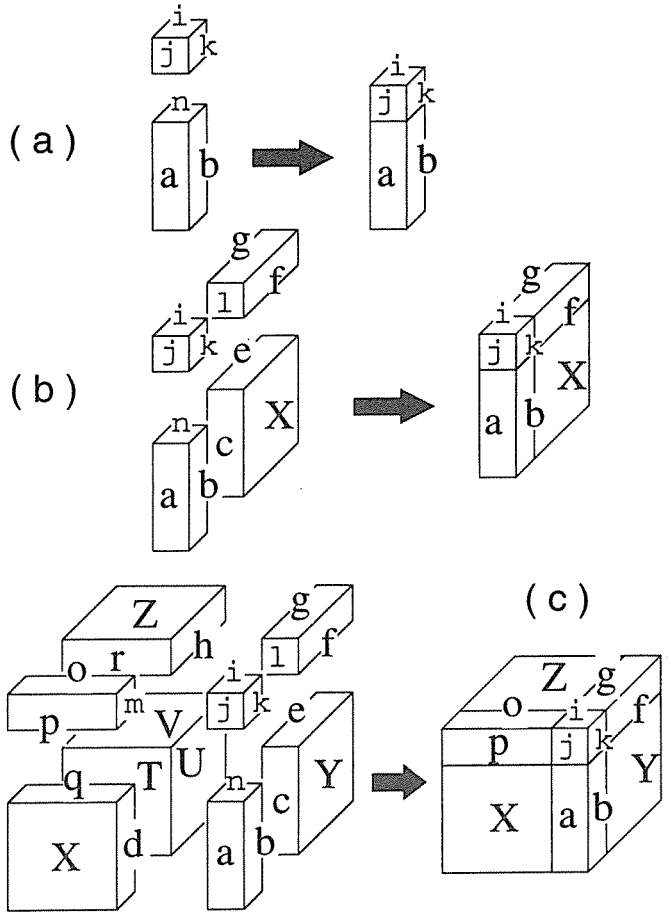

Fig. 5. Extensions of (a) P in eq. (3.2), (b) S in eq. (3.3), and (c) C in eq. (3.5).

The extended superscripts $\bar{X}$, $\bar{Y}$, and $\bar{Z}$ are defined in the same way as eq. (3.4). In eq. (3.5) we have to take care of the orientation of the surfaces T, U, and V.

Now we can express the partition function Ξ_{2N} of the cube with linear size $2N$ using the corner tensors. We first join two corner tensors (Fig. 5) to obtain a symmetric matrix

$$D^{(XU)(ZV)} = \sum_{Y} C^{XYZ}\, C^{UYV}_{\mathrm{m}}, \tag{3.6}$$

where we regard the pair (ZV) as the column index of D, and (XU) as the row index. The tensor C_{m} is the mirror image of C: $C^{UYV}_{\mathrm{m}} \equiv C^{U^TYV^T}$. The partition function Ξ_{2N} is then expressed as

$$\Xi_{2N} = \mathrm{Tr}\, D^4 = \sum_{XU} Q^{(XU)(XU)}, \tag{3.7}$$

where the matrix Q is the forth power of D (Fig. 6)

$$Q^{(XU)(ZV)} = \sum_{(AB)(CD)(EF)} D^{(XU)(AB)}\, D^{(AB)(CD)} \times D^{(CD)(EF)}\, D^{(EF)(ZV)}. \tag{3.8}$$

The matrix Q can be seen as a density matrix for the cube, because $\mathrm{Tr}\, Q$ is the partition function Ξ_{2N}. By contracting two superscripts of Q, we obtain a density submatrix

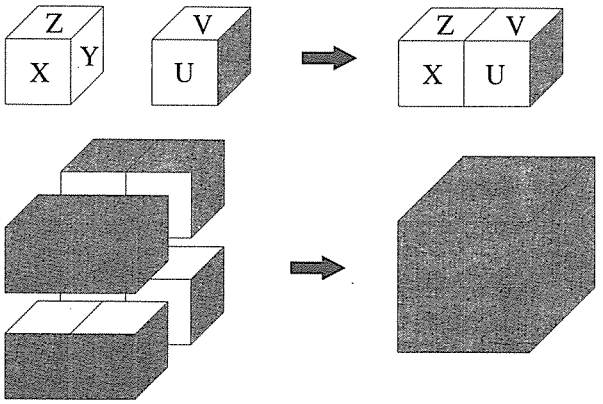

Fig. 6. The density matrix Q in eq. (3.8) is obtained by joining two corner tensors (eq. (3.6)) to obtain the tensor D, and then joining four of them. (eq. (3.8))

$$\rho^{XZ} = \sum_{U} Q^{(XU)(ZU)}, \tag{3.9}$$

which will be used for the RG transformation for the spin array.

Let us consider a density submatrix $\rho^{\bar{X}\bar{Z}}$ for the extended cube with size $2(N+1)$, where $\bar{X}$ is the extended spin array eq. (3.4); for a while we label the elements of $\bar{Z}$ as

$$\bar{Z} = \left(\begin{array}{ccc|c} x'_{11} & \cdots & x'_{1N} & f'_1 \\ \vdots & \ddots & \vdots & \vdots \\ x'_{N1} & \cdots & x'_{NN} & f'_N \\ \hline b'_1 & \cdots & b'_N & k' \end{array}\right) \tag{3.10}$$

in order to define another density submatrix. By tracing out N by $N+1$ variables of the extended density matrix $\rho^{\bar{X}\bar{Z}}$

$$\rho_{\bar{f}\bar{g}} = \sum_{b_i = b'_i \quad x_{ij} = x'_{ij}} \rho^{\bar{X}\bar{Z}}, \tag{3.11}$$

where $\bar{f} = (f_1, \ldots, f_N, k)$ and $\bar{g} = (f'_1, \ldots, f'_N, k')$, we obtain another density submatrix for the extended in-line spins. In the same way, we obtain ρ_{fg} for the in-line spins of length N - $f = (x_{1N}, \ldots, x_{NN})$ and $g = (x'_{1N}, \ldots, x'_{NN})$ - by tracing out $N-1$ by N variables of ρ^{XZ} in eq. (3.9).

§4. RG Algorithm in Three Dimension

As we have done in §2, we obtain RG transformations by way of the diagonalizations of the density submatrices. We first consider the eigenvalue relation

$$\sum_{Z} \rho^{XZ} U^{Z}_{\Psi} = \Lambda_{\Psi} U^{X}_{\Psi}, \tag{4.1}$$

where we assume the decreasing order for Λ_Ψ. We keep first m' eigenvalues, $(\Psi = 1, \ldots, m')$ and neglect the rest of relatively small ones. We then obtain the RG transformation matrix U^X_Ψ, that maps the spin array X to an m'-state block spin Ψ. For example, the corner tensor C^{XYZ} is renormalized as (Fig. 7(a))

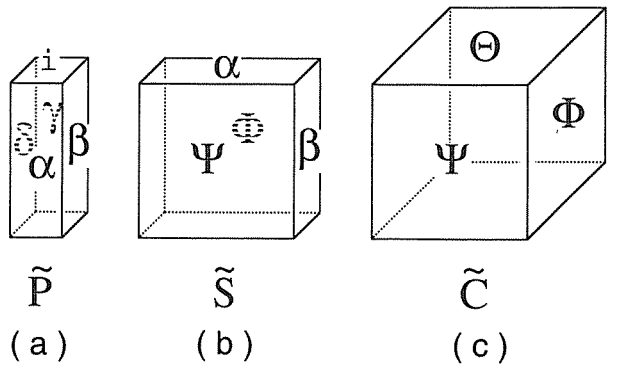

Fig. 7. The renormalized tensors (a) $\tilde{P}^{\,i}_{\alpha\beta\gamma\delta}$ in eq. (4.4), (b) $\tilde{S}^{\Psi\Phi}_{\alpha\beta}$ in eq. (4.5), and (c) $\tilde{C}^{\Psi\Phi\Theta}$ in eq. (4.2). The greek letters α, β, γ, and δ denote m-state renormalized in-line spins, and the capital ones Ψ, Φ, and Θ denote m'-state renormalized spin arrays.

$$\tilde{C}^{\Psi\Phi\Theta} = \sum_{XYZ} U^X_\Psi U^Y_\Phi U^Z_\Theta C^{XYZ}. \tag{4.2}$$

It should be noted that under the transpose of the spin array $X \to X^T$ the matrix U^X_Ψ transforms as $\pm U^X_\Psi$ according to the parity of the block spin Ψ.

Let us consider another eigenvalue relation

$$\sum_{g} \rho_{fg} A^{\psi}_{g} = \lambda^{\psi} A^{\psi}_{f} \tag{4.3}$$

for the density submatrix ρ_{fg}, where f and g are in-line spins. We assume the decreasing order for λ^ψ as before, and we keep m numbers of large eigenvalues. $(\psi = 1, \ldots, m)$ The matrix A^ψ_f then represent the RG transformation for the in-line spin f. For example, $P^{\,i}_{abcd}$ is renormalized as (Fig. 7(b))

$$\tilde{P}^{\,i}_{\alpha\beta\gamma\delta} = \sum_{abcd} P^{\,i}_{abcd} A^{\alpha}_{a} A^{\beta}_{b} A^{\gamma}_{c} A^{\delta}_{d}. \tag{4.4}$$

By using both U^X_Ψ and A^ψ_f, we can renormalize S^{XY}_{ab} as (Fig. 7(c))

$$\tilde{S}^{\Psi\Phi}_{\alpha\beta} = \sum_{abXY} S^{XY}_{ab} A^{\alpha}_{a} A^{\beta}_{b} U^X_\Psi U^Y_\Phi. \tag{4.5}$$

As a result of RG transformations, the tensors $P^{\,i}_{abcd}$, S^{XY}_{ab}, and C^{XYZ} are approximated as

$$P^{\,i}_{abcd} \sim \sum_{\alpha\beta\gamma\delta=1}^{m} A^{\alpha}_{a} A^{\beta}_{b} A^{\gamma}_{c} A^{\delta}_{d} \tilde{P}^{\,i}_{\alpha\beta\gamma\delta} \tag{4.6}$$

$$S^{XY}_{ab} \sim \sum_{\alpha\beta=1}^{m} \sum_{\Psi\Phi=1}^{m'} A^{\alpha}_{a} A^{\beta}_{b} U^X_\Psi U^Y_\Phi \tilde{S}^{\Psi\Phi}_{\alpha\beta} \tag{4.7}$$

$$C^{XYZ} \sim \sum_{\Psi\Phi\Theta=1}^{m'} U^X_\Psi U^Y_\Phi U^Z_\Theta \tilde{C}^{\Psi\Phi\Theta}. \tag{4.8}$$

For the models that have unique ground-state spin configuration, the above approximations become exact when $T = 0$ and $T = \infty$ even for $m = m' = 1$.

Now we can directly generalize the algorithm of CTMRG to 3D lattice models. The algorithm consists of the extensions for $P^{\,i}_{abcd}$ (eq. (3.2)), S^{XY}_{ab} (eq. (3.3)), and

C^{XYZ} (eq. (3.5)), and the RG transformations eqs. (4.2), (4.4) and (4.5). The procedure of the renormalization group is as follows:

(1) Start from $N = 1$, where all the tensors can be expressed by the Boltzmann weight W_{ijklmn}: $P^{i}_{abcd} = W_{iabcd\times}$, $S^{XY}_{ab} = W_{aXbY\times\times}$, and $C^{XYZ} = W_{ZXY\times\times\times}$, where the mark '$\times$' represents the fixed boundary spin.

(2) Join the tensors W_{ijklmn}, P^{i}_{abcd}, S^{XY}_{ab}, and C^{XYZ} using eqs. (3.2), (3.3), and (3.5), respectively, and obtain the extended ones $P^{i}_{\bar{a}\bar{b}\bar{c}\bar{d}}$, $S^{\bar{X}\bar{Y}}_{\bar{a}\bar{b}}$, and $C^{\bar{X}\bar{Y}\bar{Z}}$. (Increment N by one.)

(3) Using the extended corner tensor $C^{\bar{X}\bar{Y}\bar{Z}}$ in eq. (3.5), calculate the density matrix $\rho^{\bar{X}\bar{Z}}$ via eq. (3.9) and its submatrix $\rho_{\bar{f}\bar{g}}$ in eq. (3.11).

(4) Obtain the RG transformation matrix $U^{\bar{X}}_{\Psi}$ and $A^{\alpha}_{\bar{a}}$ using eqs. (4.1) and (4.3), respectively. We keep m' states for Ψ, and m states for α.

(5) Apply the RG transformations to the extended tensors to obtain $\tilde{P}^{i}_{\alpha\beta\gamma\delta}$ (eq. (4.4)), $\tilde{S}^{\Psi\Phi}_{\alpha\beta}$ (eq. (4.5)), and $\tilde{C}^{\Psi\Phi\Theta}$ (eq. (4.2)).

(6) Goto the step (2) and repeat the procedures (2)–(5) for the renormalized tensors $\tilde{P}$, $\tilde{S}$, and $\tilde{C}$.

Every time we extend the tensors in the step (2) the system size - the linear dimension of the cube - increases by 2. After the step (3) we can obtain the lower bound of the partition function by taking the trace of the density submatrix $\Xi_{2(N+1)} = \mathrm{Tr}\,\rho^{\bar{X}\bar{Z}} = \mathrm{Tr}\,\rho_{\bar{f}\bar{g}}$. We stop the iteration when the partition function per vertex converges. Since the extended spin array $\bar{X}$ of the density matrix $\rho^{\bar{X}\bar{Z}}$ contains the original (unrenormalized) spin variable, we can directly calculate the local energy and the order parameter.[29]

Let us apply the above algorithm to the 3D Ising model. The model is equivalent to the 64-vertex model whose vertex weight is given by

$$W_{ijklmn} = \sum_{\sigma=\pm1} U_{\sigma i}U_{\sigma j}U_{\sigma k}U_{\sigma l}U_{\sigma m}U_{\sigma n}, \qquad (4.9)$$

where $U_{\sigma i}$ is unity when $\sigma = i$, and is $e^{K} + \sqrt{e^{2K}-1}$ when $\sigma \neq i$. The parameter K denotes the inverse temperature $J/k_{\mathrm{B}}T$. For this model the initial conditions for step (1) are slightly modified as

$$\begin{aligned} P^{i}_{abcd} &= U_{\times i}U_{\times a}U_{\times b}U_{\times c}U_{\times d} \\ S^{XY}_{ab} &= U_{\times X}U_{\times Y}U_{\times a}U_{\times b} \\ C^{XYZ} &= U_{\times X}U_{\times Y}U_{\times Z}, \end{aligned} \qquad (4.10)$$

where '$\times$' represent the boundary Ising spin. (The modification is nothing but the change in normalizations.) We impose ferromagnetic boundary condition $\times = 1$. As a trial calculation we keep only two states for both in-line spins ($m = 2$) and spin arrays ($m' = 2$); when $m' = 2$ the parity of the renormalized spin array Ψ in eq. (4.1) is always even. Figure 8 shows the calculated spontaneous magnetization. Because of the smallness of m and m', the transition temperature T_{c} is over estimated, where the feature is common to the Klamers-Wannier approximation.[1]

Compare to the CTMRG algorithm for 2D classical systems, the above RG algorithm for 3D system requires much more computational time. The reason is that after the step (2) the extended in-line spin $\bar{f}$ becomes $2m$-state, and the extended spin array $\bar{X}$ becomes $2m^2m'$-state; in order to obtain $\rho^{\bar{X}\bar{Z}}$ in the step (3) we have to create a matrix $D^{(\bar{X}\bar{U})(\bar{Z}\bar{V})}$ by eq. (3.6), whose dimension is $4m^4m'^2$. For the simplest (non-trivial) case $m = m' = 2$ the dimension is already 256.

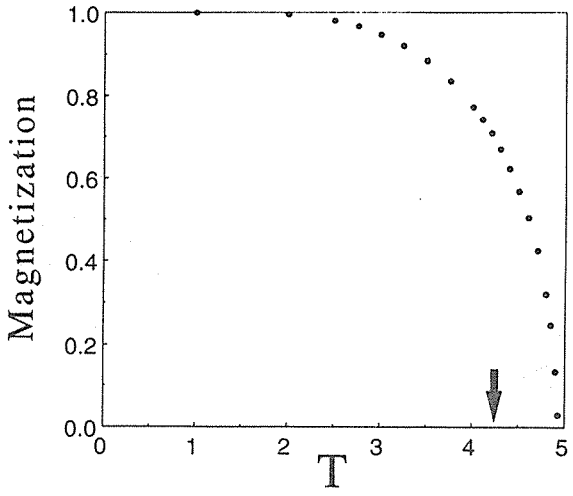

Fig. 8. Calculated spontaneous magnetization of the 3D Ising model when $m = m' = 2$. The arrow shows the true T_{c}.

§5. Conclusion and Discussion

We have explained a way of generalizing the RG algorithm of CTMRG[28,29] to 3D classical models, focusing on the construction of the density matrix from eight corner tensors. The RG transformations are obtained through the diagonalizations of the density matrices. As far as we know it is the first generalization of the infinite-system density matrix algorithm[15,16] to 3D classical systems.

From the computational view point, the calculation in 3D is far more heavy than that of CTMRG in 2D; we have to improve the numerical algorithm in 3D for realistic use. What we have done is to approximate the eigenstate of a transfer matrix in 3D as a two-dimensional product of renormalized tensor $\tilde{P}$ (eq. (4.4)); the most important process is to improve the tensor elements in $\tilde{P}$ so that the variational partition function is maximized. The improvement of tensor product state for 1D quantum system proposed by Dukelsky *et al.*,[26,27] where their algorithm does not explicitly require the density matrix, may of use to reduce the numerical effort in three dimension.

The authors would like to express their sincere thanks to Y. Akutsu, M. Kikuchi, for valuable discussions. T. N. is greatful to G. Sierra about the discussion on the matrix product state. K. O. is supported by JSPS Research Fellowships for Young Scientists. The present work is partially supported by a Grant-in-Aid from Ministry of Education, Science, Sports and Culture of Japan. Most of the numerical calculations were done by NEC SX-4 in computer center of Osaka University.

1) H. A. Kramers and G. H. Wannier: Phys. Rev. **60** (1941) 263.
2) R. Kikuchi: Phys. Rev. **81** (1951) 988.
3) H. A. Bethe: Proc. R. Soc. A **150** (1935) 552.
4) R. J. Baxter: J. Math. Phys. **9** (1968) 650 .
5) R. J. Baxter: J. Stat. Phys. **19** (1978) 461.
6) R. J. Baxter: *Exactly Solved Models in Statistical Mechanics* (Academic Press, London, 1982) p. 363.
7) N. P. Nightingale and H. W. Blöte: Phys. Rev. B **33** (1986) 659.
8) I. Affleck, T. Kennedy, E. H. Lieb and H. Tasaki: Phys. Rev. Lett. **59** (1987) 799.
9) M. Fannes, B. Nachtergale and R. F. Werner: Europhys. Lett. **10** (1989) 633.
10) M. Fannes, B. Nachtergale and R. F. Werner: Commun. Math. Phys. **144** (1992) 443.
11) M. Fannes, B. Nachtergale and R. F. Werner: Commun. Math. Phys. **174** (1995) 477.
12) A. Klümper, A. Schadschneider and J. Zittarz: Z. Phys. B **87** (1992) 281.
13) H. Niggemann, A. Klümper and J. Zittartz: Z. Phys. B **104** (1997) 103.
14) H. Takasaki and T. Nishino: unpublished.
15) S. R. White: Phys. Rev. Lett. **69** (1992) 2863.
16) S. R. White: Phys. Rev. B **48** (1993) 10345.
17) S. Östlund and S. Rommer: Phys. Rev. Lett. **75** (1995) 3537.
18) S. Rommer and S. Östlund: Phys. Rev. B **55** (1997) 2164.
19) S. R. White and D. A. Huse: Phys. Rev. B **48** (1993) 3844.
20) K. Hida: J. Phys. Soc. Jpn. **65** (1996) 895.
21) G. Sierra and M. A. Martín-Delgado: *Strongly Correlated Magnetic and Superconducting Systems* (Springer Berlin, 1997), and references there in.
22) T. Nishino: J. Phys. Soc. Jpn. **64** (1995) 3598.
23) Enrico Carlon and Andrej Drzewiński: Phys. Rev. Lett. **79** (1997) 1591.
24) Enrico Carlon and Andrej Drzewiński: cond-mat/9709176.
25) G. Sierra, M. A. Martín-Delgado, J. Dukelsky, S. R. White and D. J. Scalapino: cond-mat/9707335.
26) J. Dukelsky, M. A. Martín-Delgado, T. Nishino and G. Sierra: cond-mat/9710310.
27) J. M. Roman, G. Sierra, J. Dukelski and M. A. Martín-Delgado: cond-mat/9802150.
28) T. Nishino and K. Okunishi: J. Phys. Soc. Jpn. **65** (1996) 891.
29) T. Nishino and K. Okunishi: J. Phys. Soc. Jpn. **66** (1997) 3040.
30) T. Nishino and K. Okunishi: J. Phys. Soc. Jpn. **67** (1998) 1492.
31) The generalization of CTMRG to asymmetric vertex models is straight forward.
32) To introduce such a symmetry is not a over simplification. For example, the $d = 3$ Ising model can be expressed as a three-dimensional 64-vertex model. (See eqs. (4.9)–(4.12).)

SX-4