High Performance Computing for Eigenvalue Solver in Density-Matrix Renormalization Group Method: Parallelization of the Hamiltonian matrix-vector multiplication

Susumu Yamada, Masahiko Okumura, and Masahiko Machida

CCSE, Japan Atomic Energy Agency, 6-9-3 Higashi-Ueno,
Taito-ku, Tokyo, 110-0015, Japan
&
CREST(JST), 4-1-8 Honcho, Kawaguchi, 330-0012, Japan
{yamada.susumu,okumura.masahiko,machida.masahiko}@jaea.go.jp

Abstract. The Density Matrix Renormalization Group (DMRG) method is widely used by computational physicists as a high accuracy tool to obtain the ground state of large quantum lattice models. Since the DMRG method has been originally developed for 1-D models, many extended method to a 2-D model have been proposed. However, some of them have issues in term of their accuracy. It is expected that the accuracy of the DMRG method extended directly to 2-D models is excellent. The direct extension DMRG method demands an enormous memory space. Therefore, we parallelize the matrix-vector multiplication in iterative methods for solving the eigenvalue problem, which is the most time- and memory-consuming operation. We find that the parallel efficiency of the direct extension DMRG method shows a good one as the number of states kept increases.

Key words: parallel and distributed computing, DMRG method, matrix-vector multiplication, eigenvalue problem, quantum lattice systems

1 Introduction

Quantum lattice systems, e.g. Heisenberg model[1] and Hubbard model[2, 3], have attracted a tremendous number of physicists since the systems exhibit a lot of interesting phenomenon such as High-T_c superconductivity. In order to understand the systems, some computational methods have been proposed. The most accurate one of them is the exact diagonalization method, which solves the ground state (the smallest eigenvalue and the corresponding eigenvector) of the Hamiltonian matrix derived from the systems. We have actually parallelized the exact diagonalization method and obtained some novel physical results[4–7]. However, the dimension of the Hamiltonian matrix for the exact diagonalization method increases almost exponentially with the number of the lattice sites. Thus, the limit of the simulation on a supercomputer with a terabyte memory system is an about-20-site system[6, 7].

In order to overcome the memory-size explosion problem, the Density Matrix Renormalization Group (DMRG) method, which keeps the number of the relevant quantum states constant by renormalizing the states of the previous step on enlarging the system (see Fig. 1), has been proposed by S. R. White[8, 9]. The DMRG has originally been developed for 1-leg (1-D) lattice models, and a lot of problems for 1-D quantum lattice systems has been resolved. The DMRG method can be directly extended to an s-leg (2-D) model as depicted in Fig. 2. The extension strategy is promising for the excellent accuracy and the good convergence property. However, the method leads to a large amount of memory consumption, since the maximum number of the states required in the direct extension DMRG algorithm is given as $16^s m^2$ for the s-leg Hubbard model per block, in which s and m are the number of the sites in the rung direction and the number of states kept, respectively. Although the degree of freedom practically decreases by eliminating physically irrelevant states, it is clear that even a slight increment of the legs gives rise to an exponential growth of the state number. Therefore, the previous 2-D DMRG method have adopted "multichain algorithm", as depicted in Fig. 3, since its memory space is principally comparable to the 1-D case. However, there remain unsolved issues in terms of its accuracy[10, 11]. Thus, we study the parallelization technique for the direct extension of DMRG method to s-leg models on a distributed-memory parallel computer, which totally has a huge memory system.

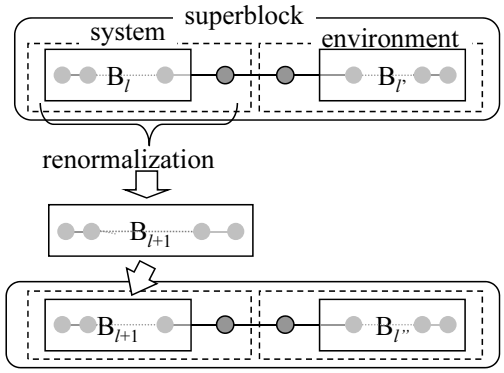

Fig. 1. A schematic figure of the renormalization scheme of DMRG method for a 1-D lattice model. The "superblock" is composed of the "system" and the "environment". The rectangles inside of the above superblock indicate the blocks which have l and l' lattice sites, and the dark circles present single sites. New block B_{l+1} is formed by renormalizing the left block B_l and the left single with keeping the number of the states in the system block.

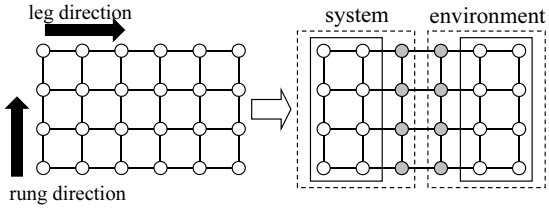

Fig. 2. A superblock configuration in the direct extension of DMRG to a 4-leg model.

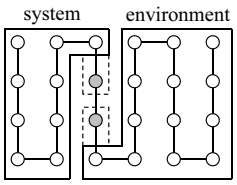

Fig. 3. A superblock in the multichain algorithm.

2 Density matrix renormalization group method

2.1 Algorithm

In the DMRG method, the whole system of the lattice model is called "superblock", and the "superblock" is split into two blocks, which are called "system block" and "environment block" (see Fig. 1). The procedure of the DMRG method for 1-D model is described as follows:

1. Form the Hamiltonian matrix form the superblock and find the ground state of the matrix.
2. Form the density matrix for the system block using the ground state.
3. Solve the eigenvalue problem for the density matrix, and discard all expect for the largest m eigenvalues and the corresponding eigenvectors. Here, m means the number of states kept per block.
4. Form the new system block, which is the previous system block plus one site, using the eigenvectors, and construct the new superblock out of the new system block and the relevant environment block.

In the direct extension DMRG for the s-leg model, a new system block is formed with the previous system block and s sites. In this case, the dimension of the Hubbard Hamiltonian matrix for the new superblock is $16^s m^2$. Although the dimension can decrease by eliminating the irrelevant states, the calculation for solving the ground state of the Hamiltonian matrix is the most time- and memory-consuming operation. Since the Hamiltonian matrix is a sparse symmetric matrix, an iterative method, such as the Lanczos method [12] and the conjugate gradient method [13, 14], are utilized for solving the ground state. For these iterative methods, the most difficult part is the parallelization of the multiplication of the Hamiltonian matrix and a vector. Therefore, we focus on the parallelization for the multiplication in the following.

2.2 Parallelization of matrix-vector multiplication in DMRG

Each block of the superblock for the direct extension 2-D DMRG method is called "block 1", "block 2", "block 3", and "block 4" from the left, and state

of the "block j" is represented as i_j. We decompose the superblock into the left block, the central block, and the right block as Fig. 4, and the Hamiltonian matrices for the three blocks are represented as H_l, H_c, and H_r, respectively. The Hamiltonian matrix for the superblock is decomposed into three matrices, i.e. H_1, H_2, and H_3, which correspond to the states i_1 and i_2, i_3 and i_4, and i_2 and i_3, respectively. Thus, the matrix-vector multiplication Hv is partitioned as

$$Hv = H_1 v + H_2 v + H_3 v. \tag{1}$$

Here, let the $((i_3 - 1)m^2 n + (i_4 - 1)mn + (i_2 - 1)m + i_1)$-th element of the vector v, which corresponds to the $|i_1 i_2 i_3 i_4 >$ state, transform into the element $((i_2 - 1)n + i_1, (i_4 - 1)m + i_3)$ of a matrix V (see Fig. 5), in which $n = 4^s$ in s-leg ladder Hubbard model. Then, the first two multiplications of (1) can be transformed into matrix-matrix multiplications as

$$H_1 v \rightarrow H_l V,$$
$$H_2 v \rightarrow V H_r^T.$$

In addition, when the element of the vector v is transformed into the element $((i_3 - 1)n + i_2, (i_4 - 1)n + i_1))$ of a matrix V_c (see Fig. 5), the last operation of (1) is also transformed into a matrix-matrix multiplication as

$$H_3 v \rightarrow H_c V_c.$$

We find that the matrix-vector multiplication Hv can be decomposed into three matrix-matrix multiplications. It is expected that the multiplications can be effectively parallelized by partitioning the matrices V and V_c. The parallel computation for the multiplication Hv is realized by four calculation stages and four communication ones as follows:

CAL 1: $W_1^c = H_l V^c$,
COM 1: communication for transforming V^c to V^r,
CAL 2: $Z_2^r = V^r H_r^T$,
COM 2: communication for transforming Z_2^r to W_2^c,
COM 3: communication for transforming V^c to V_c^c,
CAL 3: $Z_3^c = H_c V_c^c$
COM 4: communication for transforming Z_3^c to W_3^c,
CAL 4: $W^c = W_1^c + W_2^c + W_3^c$,

where the superscription c and r denote the columnwise and rowwise partitioning in matrix data distribution, respectively. Although the matrices V and V_c are originally dense matrices, the elements corresponding to the irrelevant states are eliminated. Therefore, the matrices should be partitioned in consideration of the arrangement of the relevant elements.

2.3 Data locality

In the Hubbard model, there is no correlation among states with different number of spins. Therefore, the three matrices H_l, H_c, and H_r, which are the Hamiltonian

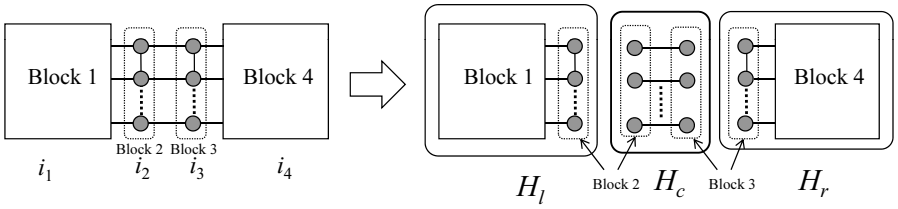

Fig. 4. A definition for the Hamiltonian matrices H_l, H_r, and H_c. The superblock as shown in the left side is decomposed into three blocks, i.e. the left block, the right block, and central block, as shown in the right side. The matrix H_l, H_r, and H_c are the Hamiltonian matrices for the left block, the right block, and central block, respectively.

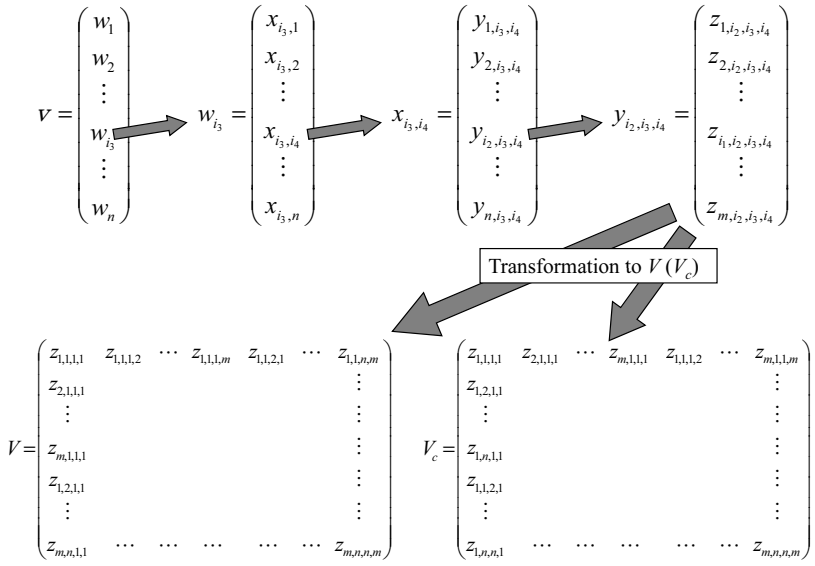

Fig. 5. A way to transform the vector v to the matrices V and V_c. The vector v is hierarchically arranged as the upper of the figure. The element $z_{i_1 i_2 i_3 i_4}$is arranged in the matrix as this figure.

matrices for the left block, central block, and right block, respectively, become the block diagonal matrices, if the elements are rearranged by the number of spins. The rearrangement improves the data locality, which strongly influences the performance on a scalar computer.

3 Numerical experiment

We examine the performance of the direct extension 2-D DMRG method on SGI Altix3700Bx2 in Japan Atomic Energy Agency. A test example is 4-leg (10×4-site) Hubbard model with 38 fermions (19 up-spins, 19 down-spins). Figure 6 shows a relationship between the number of states kept m and the elapsed time. We find that as the number of processors increases from 32 to 128, the elapsed time is reduced to be about one half, when m is about larger than 128. The result shows that the present parallelization scheme is promising, since the obtained ground state approaches to the true ground state with increasing the number of the states kept m.

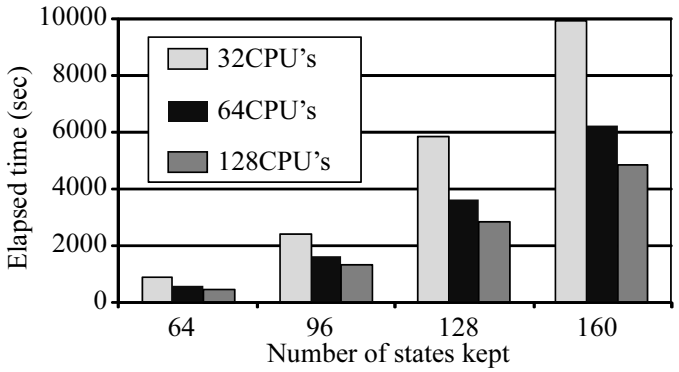

Fig. 6. Relationship between the number m of states kept and the elapsed time of the direct extension DMRG method for 10×4-site Hubbard model on SGI Altix 3700Bx2.

4 Conclusion

In order to execute the direct extension of DMRG method to s-leg model on distributed-memory parallel computers, we parallelized and tuned mainly the matrix-vector multiplication in solving the eigenvalue problem for the Hamiltonian matrix, which is the most time- and memory-consuming operation. Consequently, we found that the parallelization efficiency increases with increasing the number of the states kept. Therefore, we believe that the direct extension 2-D DMRG method is a promising tool to explore large quantum 2-D lattice systems.

References

1. T. Kennedy and B. Nachtergaele, The Heisenberg Model - a Bibliography (1996). http://math.arizona.edu/~tgk/qs.html
2. M. Rasetti, ed., The Hubbard Model: Recent Results, World Scientific, Singapore (1991)
3. A. Montorsi, ed., The Hubbard Model, World Scientific, Singapore (1992)
4. M. Machida, S. Yamada, Y. Ohashi, and H. Matsumoto, Novel superfluidity in a trapped gas of Fermi atoms with repulsive interaction loaded on an optical lattice, *Phys. Rev. Lett.*, 93, 200402 (2004)
5. M. Machida, S. Yamada, Y. Ohashi, and H. Matsumoto, Machida et al. Reply, *Phys. Rev. Lett.*, 95, 218902 (2005)
6. S. Yamada, T. Imamura, and M. Machida, 16.447 TFlops and 159-Billion-dimensional Exact-diagonalization for Trapped Fermion-Hubbard Model on the Earth Simulator, *Proc. of SC05* (2005) http://sc05.supercomputing.org/schedule/pdf/pap188.pdf
7. S. Yamada, T. Imamura, T. Kano, and M. Machida, High-Performance Computing for Exact Numerical Approaches to Quantum Many-Body Problems on the Earth Simulator, *Proc. of SC06* (2006) http://sc06.supercomputing.org/schedule/pdf/gb113.pdf
8. S. R. White, Density Matrix Formulation for Quantum Renormalization Groups, Phys. Rev. Lett. 69, 2863-2866 (1992)
9. S. R. White, Density-matrix algorithms for quantum renormalization groups, Phys. Rev. B 48, 10345-10355 (1993)
10. G. Hager, G. Wellein, E. Jackemann, and, H. Fehske, Stripe formation in dropped Hubbard ladders, Phys. Rev. B, 71, 075108 (2005)
11. R. M. Noack and S. R. Manmana. Diagonalization and Numerical Renormalization-Group-Based Methods for Interacting Quantum Systems, Proc. of AIP Conf., 789, 91-163 (2005)
12. J.K. Cullum and R.A. Willoughby, Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol.1: Theory (Philadelphia: SIAM, (2002)
13. A. V. Knyazev, Preconditioned eigensolvers - An oxymoron?, Electronic Transactions on Numerical analysis 7, 104-123 (1998)
14. A. V. Knyazev, Toward the optimal eigensolver: Locally optimal block preconditioned conjugate gradient method, SIAM J. Sci. Comput., 23, 517-541 (2001)