Journal of the Earth Simulator, Volume 7, June 2007, 23−35

Ultra Large-scale Exact-diagonalization for Confined Fermion-Hubbard Model on the Earth Simulator: Exploration of Superfluidity in Confined Strongly-Correlated Systems

Susumu Yamada[1], Toshiyuki Imamura[2], Takuma Kano[1],
Yoji Ohashi[3], Hideki Matsumoto[4], and Masahiko Machida[1]*

[1] *Center for Computational Science and e-Systems, Japan Atomic Energy Agency, 6-9-3, Higashi-Ueno, Taito-ku, Tokyo, 110-0015, Japan*
[2] *Department of Computer Science, The University of Electro-Communications, 1-5-1, Chofugaoka, Chofu, Tokyo, 182-8585, Japan*
[3] *Faculty of Science and Technology, Keio University, 3-14-1, Hiyoshi, Kohoku-ku, Yokohama, Kanagawa, 223-8522, Japan*
[4] *Institute of Physics, University of Tsukuba, 1-1-1, Tennodai, Tsukuba, Ibaraki, 305-8573, Japan*

CREST(JST), 4-1-8, Honcho, Kawaguchi, Saitama, 332-0012, Japan

(Received January 12, 2007; Revised manuscript accepted March 6, 2007)

Abstract In order to explore a possibility of superfluidity in confined strongly-correlated fermion systems, e.g., nano-scale cuprate High-Tc superconductors and atomic Fermi gases loaded on optical lattice, we implement an exact diagonalization code for their mathematical model, i.e., a trapped Hubbard model on the Earth Simulator. We compare two diagonalization algorithms, the traditional Lanczos method and a new algorithm, the preconditioned conjugate gradient (PCG) method, and find that when using the PCG the total CPU time can be reduced to 1/3 ~ 1/5 compared to the former one since the convergence can be dramatically improved by choosing a good preconditioner and the communication overhead is much more efficiently concealed in the PCG method. Consequently, such a performance improvement enables us to do systematic studies for several parameters. Numerical simulation results reveal that an unconventional type of pairing specific to the confined system, which may cause superfluidity, develops under a strong repulsive interaction.

Keywords: superfluidity, exact-diagonalization, Hubbard model, atom Fermi gas, preconditioned conjugate gradient method

1. Introduction

The successful achievements [1, 2, 3] of the Bose-Einstein condensation in the trapped atomic Bose gas were honored by the Nobel Prize in 2001. After that, atomic physicists have challenged another more difficult condensation in the atomic Fermi gas [4, 5]. The condensation and the resultant superfluidity in fermion system is one of the most universal issues in fundamental physics, since particles which form matters, i.e., electron, proton, neutron, quark, and so on, are fermions and an exploration for their many-body ground states is a central target in modern physics. Motivated by interests based on such a wide background, we numerically explore a possibility of superfluidity in the trapped atomic Fermi gases [6]. Our undertaking model is the fermion-Hubbard model [7] with a trapping potential. The Hubbard model describes a many-body fermion system on a discrete lattice, and the model captures an essence of strongly-correlated electronic structures in solid state systems. Furthermore, whether the model can describe the high temperature superconductivity or not is a main issue in condensed matter physics. In atomic gases, the model system can be realized by a standing wave created due to two laser interference [8] in the atomic Fermi gas.

The Hubbard model is one of the most intensively-studied models by computers because it owns very rich physics, although the model expression is quite simple [7]. The Hamiltonian of the Hubbard model with a trap

* ***Corresponding author***: *Masahiko Machida, Center for Computational Science and e-Systems, Japan Atomic Energy Agency, 6-9-3, Higashi-Ueno, Taito-ku, Tokyo, 110-0015, Japan. E-mail: machida.masahiko@jaea.go.jp*

potential [6, 9] is given as

$$H = -t \sum_{i,j,\sigma} (a^{\dagger}_{j\sigma} a_{i\sigma} + H.C.) + U \sum_{i} n_{i\uparrow} n_{i\downarrow} + \left(\frac{2}{N}\right)^2 V \sum_{i,\sigma} n_{i\sigma} \left(i - \frac{N}{2}\right)^2 , \quad (1)$$

where t, U, and V are the hopping parameter from i-th to j-th sites (normally j is the nearest neighbor site of i), the repulsive energy for on-site double occupation of two fermions, and the parameter characterizing the strength of the trapping potential, respectively, as schematically shown in Fig. 1, and $a_{i,\sigma}$, $a^{\dagger}_{i,\sigma}$ and $n_{i,\sigma}$ are the annihilation, the creation, and the number operator of a fermion with pseudo-spin σ (= $\uparrow$ (up) or $\downarrow$ (down)) on the i-th site, respectively.

The computational finite-size approaches on the Hubbard model are roughly classified into three types. The first one is the exact diagonalization using the Lanczos method [10], the second one is the quantum Monte Carlo [7], and the third one is the density matrix renormalization group (DMRG) method. The first one directly calculates the ground and the low lying excited states of the model, and moreover, obtains various physical quantities with considerably high accuracy. However, the numbers of fermions and sites are severely limited because the matrix size of the Hubbard Hamiltonian approximately grows exponentially with increasing these numbers (see Appendix A for how to make a matrix and Table 3 for an example about how the matrix size depends on the numbers). On the other hand, the second and third ones have a great advantage in these numbers, but include specific problems to be solved. The Monte Calro method confronts a fatal problem as the negative sign in the probability calculation [7], and therefore, its reliability has been always argued. The third one, the DMRG scheme has been regarded as an alternative way to approach the ground state in a very high accuracy like the exact diagonalization, but the application is limited to 1-D or ladder system. From these contexts, if infinite computational resources are permitted, the exact diagonalization is clearly the best approach. Thus, we challenge a theme for supercomputing, that is, to implement the exact diagonalization code on the present top-class supercomputer, i.e., the Earth Simulator [11], and to examine how large matrices can be solved and how excellent performance can be obtained. Also, we note that the matrix diagonalization issue is one of the most central ones in the so-called high performance computing (HPC) because the matrix operation is the most basic one.

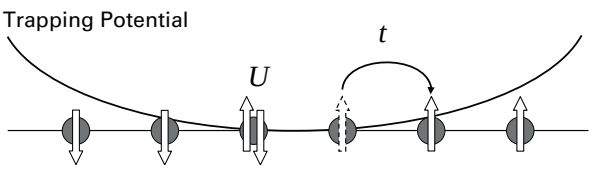

Fig. 1 A schematic figure of the one-dimensional fermion-Hubbard model with a trapping potential, where t and U are the hopping parameter and the repulsive energy in the double occupation on a site, respectively. The up-arrow and the down-arrow stand for fermion with up and down (pseudo) spin, respectively.

The Earth Simulator developed by NASDA (presently JAXA), JAERI (presently JAEA), and JAMSTEC, is situated on the flagship class of highly parallel vector supercomputer of the distributed-memory type. The theoretical peak performance is 40.96 TFlops, and the total memory size is 10.24 TB. The architecture is quite suitable especially for scientific and technological large-scale computations [11], due to well-balance of the processing speed of the floating point operation and the memory bandwidth as well as the network throughput. Therefore, several applications achieved excellent performance, and some of them won the Gordon Bell Prize at the *Supercomputing* conferences [12, 13, 14, 15, 16]. Furthermore, we would like to note that our papers about HPC issues on the Earth Simulator was chosen as finalists of the Gordon Bell Prize even at *Supercomputing* in 2006 year (see the website[17] and reference[18]), although its peak performance is listed below the top 10 in 2006. This is an direct evidence how the Earth Simulator is an excellent machine for scientific large-scale simulations.

In this paper, we develop a new type of high performance application which solves the eigenvalue problem of the Hubbard Hamiltonian matrix (1) on the Earth Simulator and present our progress in numerical algorithm and parallelization technique to obtain the best performance and solve the world-record class of large matrices. In the algorithmic issue, we suggest a new profitable algorithm, i.e., the preconditioned conjugate gradient (PCG) method[1] as an alternative one for the huge-scale matrix diagonalization. Comparing between the PCG method and the conventional Lanczos method, we find that the PCG method is much more excellent than the Lanczos method except for the memory usage. On the other hand, in the parallelization technique issue, we examine an effect of the hybrid parallelization which combines the inter-node parallelization using MPI for dis-

[1] Calculating the smallest eigenvalue of the matrix H is equivalent to the minimization of the Rayleigh quotient, and the minimization is executed by using the PCG method.

tributed memory with the intra-node parallelization using the automatic parallelization for shared memory in addition to the vectorization. Consequently, we find a practical technique to save the memory by using the hybrid parallelization. Such a technique is crucial when efficient memory use is severely required.

The contents of this paper are as follows. In Section 2, we introduce two eigenvalue solvers to diagonalize the Hamiltonian matrix of the Hubbard model and compare their convergence properties. Section 3 presents the parallelization technique of two solvers on the Earth Simulator, and especially focus on a specific technique to save the memory. Section 4 shows actual performance in large-scale matrix diagonalizations on the Earth Simulator. Finally, we present typical simulation results which show a possibility of an unconventional superfluidity in the Hubbard model with the trap potential in Section 5.

2. Numerical Algorithms

The core routine of the target program calculates the smallest eigenvalue and the corresponding eigenvector for $H\upsilon = \lambda\upsilon$, where Hamiltonian matrix H is sparse, real, and symmetric (see the Appendix A for the matrix character). Thus, several iterative numerical algorithms, i.e., the power method, the Lanczos method, the conjugate gradient (CG) method, and so on, are applicable. In the following, we concentrate on two numerical algorithms, the Lanczos method and the PCG method, from viewpoints of the memory usage and the performance on the Earth Simulator.

$\boldsymbol{x}_0 :=$ an initial guess.
$\beta_0 := 1, \boldsymbol{v}_{-1} := 0, \boldsymbol{v}_0 := \boldsymbol{x}_0/\|\boldsymbol{x}_0\|$
do i=0,1,... $m-1$, or until $\beta_i < \epsilon$
  $\boldsymbol{u}_i := H\boldsymbol{v}_i - \beta_i \boldsymbol{v}_{i-1}$
  $\alpha_i := (\boldsymbol{u}_i, \boldsymbol{v}_k)$
  $\boldsymbol{w}_{i+1} := \boldsymbol{u}_i - \alpha_i \boldsymbol{v}_i$
  $\beta_{i+1} := \|\boldsymbol{w}_i\|$
  $\boldsymbol{v}_{i+1} := \boldsymbol{w}_i/\beta_{i+1}$
enddo

(a) Lanczos method

$\boldsymbol{x}_0 :=$ an initial guess, $\boldsymbol{p}_0 := 0$
$\boldsymbol{x}_0 := \boldsymbol{x}_0/\|\boldsymbol{x}_0\|$, $X_0 := H\boldsymbol{x}_0$, $P_0 = 0$, $\mu_{-1} := (\boldsymbol{x}_0, X_0)$
$\boldsymbol{w}_0 := X_0 - \mu_{-1}\boldsymbol{x}_0$
do k=0, ... until convergence
  $W_k := H\boldsymbol{w}_k$
  $S_A := \{\boldsymbol{w}_k, \boldsymbol{x}_k, \boldsymbol{p}_k\}^T\{W_k, X_k, P_k\}$
  $S_B := \{\boldsymbol{w}_k, \boldsymbol{x}_k, \boldsymbol{p}_k\}^T\{\boldsymbol{w}_k, \boldsymbol{x}_k, \boldsymbol{p}_k\}$
  Solve the smallest eigenvalue μ
    and the corresponding vector $\boldsymbol{v}$,
    $S_A\boldsymbol{v} = \mu S_B\boldsymbol{v}$, $\boldsymbol{v} = (\alpha, \beta, \gamma)^T$.
  $\mu_k := (\mu + (\boldsymbol{x}_k, X_k))/2$
  $\boldsymbol{x}_{k+1} := \alpha\boldsymbol{w}_k + \beta\boldsymbol{x}_k + \gamma\boldsymbol{p}_k$, $\boldsymbol{x}_{k+1} := \boldsymbol{x}_{k+1}/\|\boldsymbol{x}_{k+1}\|$
  $\boldsymbol{p}_{k+1} := \alpha\boldsymbol{w}_k + \gamma\boldsymbol{p}_k$, $\boldsymbol{p}_{k+1} := \boldsymbol{p}_{k+1}/\|\boldsymbol{p}_{k+1}\|$
  $X_{k+1} := \alpha W_k + \beta X_k + \gamma P_k$, $X_{k+1} := X_{k+1}/\|\boldsymbol{x}_{k+1}\|$
  $P_{k+1} := \alpha W_k + \gamma P_k$, $P_{k+1} := P_{k+1}/\|\boldsymbol{p}_{k+1}\|$
  $\boldsymbol{w}_{k+1} := T(X_{k+1} - \mu_k\boldsymbol{x}_{k+1})$, $\boldsymbol{w}_{k+1} := \boldsymbol{w}_{k+1}/\|\boldsymbol{w}_{k+1}\|$
enddo

(b) Preconditioned conjugate gradient method

Fig. 2 Algorithms of two eigenvalue solvers.

2.1 Lanczos Method

The Hubbard Hamiltonian Eq.(1) is represented by a large sparse and symmetric matrix. Thus, the Lanczos method has been traditionally employed to save the memory space and calculate the matrix whose size is as large as possible. The algorithm of the Lanczos method is summarized in the Fig. 2(a).

The main recurrence part of this algorithm repeats to update the Lanczos vector υ_{i+1} from υ_{i-1} and υ_i as seen in Fig. 2(a). Therefore, the memory requirement of the recursion is $2N$ words. In addition, an N-word buffer is required for storing an eigenvector. Consequently, the total memory requirement of the Lanczos method is $3N$ words.

As the algorithm is shown in Fig. 2(a), the main loop iterates until m or the coefficient $\beta_i < \varepsilon$. Thus, the upper limit of the iterations is formally m. Some reports point out that m should be set $2\sqrt{N}$ ordinarily. However, according to several preliminary tests, $2\sqrt{N}$ is too large to approximate the maximum eigenvalue in our case. In the following, we choose much smaller empirical number than $2\sqrt{N}$, which is N-independent number, e.g., 200 or 300, as an iteration count.

2.2 Conjugate Gradient Method

Since the CPU time generally grows as the matrix size increases, the performance improvement is a crucial factor for systematic studies repeating huge matrix diagonalization. Thus, we have suggested an alternative algorithm based on the conjugate gradient (CG) theory. Among various CG methods, we employed an algorithm proposed by Knyazev[22, 23], which searches for the minimum eigenvector with use of the direction vector calculated by Ritz vector as shown in Fig. 2(b).

The algorithm requires a memory space to store six vectors, i.e., the residual vector ω_i, the search direction vector p_i, and the eigenvector x_i, and moreover, W_i, P_i, and X_i which are matrixvector products $H\omega_i$, Hp_i, and Hx_i, respectively. Thus, the memory usage is totally $6N$ words, which is two times larger than that of the Lanczos method. In the algorithm depicted in Fig. 2(b), an operator T indicates the preconditioner, whose choice is crucial for fast convergence. In the present Hubbard-hamiltonian matrix, we find that the zero-shift point Jacobi preconditioner is the best one from the convergence test. Details of the preconditioning are given in Appendix B [24].

2.3 Performance Test of Two Algorithms

Two algorithms, the Lanczos method and the PCG method, are compared in terms of the memory usage, the

number of iterations, the elapsed time, and the performance. In addition to the basic memory requirement described in Section 2.1 and 2.2, both algorithms demand an N-word buffer for diagonal elements of the Hamiltonian matrix and additionally two N-word buffers (see Section 3.2 and 3.3) to execute parallel calculations.

Table 1 summarizes the total memory usage, the number of iterations, the elapsed time, and the performance (Flops rate) for an eigenvalue calculation of a 1,502,337,600-dimensional Hamiltonian matrix (12 fermions on 20 sites) by using 10 nodes of the Earth Simulator (80 processor elements). The result illustrates that the PCG method is an overwhelmingly powerful algorithm except for the memory requirement. In contrast to the Lanczos method with fixed iteration counts, the preparation of well-approximated initial eigenvalues or vectors can dramatically reduce the iteration counts in the PCG method. Thus, we adopt the PCG method in most simulations, while we use the Lanczos method only in cases of the ultra large-scale problem in which the PCG method is not applicable. In the following Section 3, we show parallelization techniques common for the two methods, and in Section 4 we compare the two methods from viewpoints of the memory usage and the performance on real large-scale eigenvalue problems whose dimensions are over a hundred billion.

Table 1 A performance test of two algorithms on 10 nodes of the Earth Simulator for a problem of 12 fermions on 20 sites.

	Lanczos	PCG
Memory Requirement	6N	9N
[Byte]	67.2G	100.7G
Iteration Controllability	Fixed	Variable
# Iterations	200	91
Residual Error	8.3523E-9	1.255E-9
Elapsed Time [sec]	95.0	28.2
Flops	269.5G	391.4G
(Peak Ratio)	(42.1%)	(61.1%)

3. Parallelization on the Earth Simulator

A node of the Earth Simulator is composed of eight vector PE's and its memory is shared among them. In order to achieve high performance in such an architecture, the intra-node parallelism is crucial as well as the inter-node parallelization. In the intra-node parallel programming, we adopt the automatic parallelization of the compiler system using a special language extension, so-called CDIR compiler directive. In the inter-node parallelization, we utilize the MPI library tuned for the Earth Simulator.

In this section, we focus on a core operation $H\boldsymbol{v}$ common for both the Lanczos and the PCG algorithms and present the parallelization including data partitioning, the communication, and the overlap. Furthermore, we give two technical remarks, i.e., a technique to save the memory by combining inter and intra-node parallelizations and an effective technique for the usage of vector pipelines.

3.1 Core Operation: Matrix-Vector Multiplication

In both the Lanczos and PCG methods (see Fig. 2(a) and (b)), a core calculation is $H\boldsymbol{v}$, where H is the Hamiltonian matrix and $\boldsymbol{v}$ is an eigenvector. The both methods are composed of several times repetition of the core calculation. Therefore, we focus on the numerical algorithm and the parallelization scheme on $H\boldsymbol{v}$. By using a matrix representation, the Hubbard Hamiltonian H (1) is mathematically given as

$$H = I \otimes A + A \otimes I + D, \tag{2}$$

where I, A, and D are the identity matrix, the sparse matrix due to the hopping between neighboring sites, and the diagonal matrix originated from the presence of the on-site repulsion, respectively. We note that in large-scale problems the non-zero elements of large-matrices $I \otimes A$ and $A \otimes I$ can not be expanded on the whole memory, while A and D can be stored on the memory of each node.

In the core operation $H\boldsymbol{v}$, the matrix-vector multiplications are transformed into the matrix-matrix multiplications as

$$H\boldsymbol{v} \mapsto \begin{cases} D\boldsymbol{v} & \mapsto \bar{D} \odot V \\ (I \otimes A)\boldsymbol{v} & \mapsto AV \\ (A \otimes I)\boldsymbol{v} & \mapsto VA^T \end{cases} \tag{3}$$

where the matrix V is constituted from the vector $\boldsymbol{v}$ by the following procedure. First, decompose the vector $\boldsymbol{v}$ into n blocks, and rearrange in the two-dimensional manner as follows,

$$\boldsymbol{v} = (\underbrace{v_{1,1}, v_{2,1}, \ldots, v_{n,1}}_{\text{first block}}, \underbrace{v_{1,2}, v_{2,2}, \ldots, v_{n,2}}_{\text{second block}}, \ldots, \underbrace{v_{1,n}, v_{2,n}, \ldots, v_{n,n}}_{n\text{-th block}})^T.$$

Here, a pair of subscripts of each element $\boldsymbol{v}$ formally indicates a position of row and column of the matrix V. The k-th element of the matrix D, d_k, is also mapped onto the matrix $\bar{D}$ in the same manner described above, and the operator $\odot$ means an elementwise multiplication.

3.2 Data Distribution, Parallel Calculation, and Communication

The matrix A, which represents the site hopping of up (or down) spin fermions, is a sparse matrix, and its sparsi-

ty is high. In contrast, the matrices V and $\overline{D}$ must be treated as dense matrices, and their dimension is large enough to overfull the memory capacity of a single node. Therefore, all nonzero elements of the matrix A are stored on all the nodes in the compressed row storage format, while the matrices V and $\overline{D}$ are columnwisely partitioned. On the other hand, the rowwisely partitioned V must be also stored on each node for parallel computing of VA^T.

The core operation $H\boldsymbol{v}$ including the data communication are composed of six stages as follows:

CAL1: $E^c = \overline{D}^c \odot V^c$,
CAL2: $W_1^c = E^c + AV^c$,
COM1: communicatin to transpose V^c into V^r,
CAL3: $W_2^r = V^r A^T$,
COM2: communicatin to transpose W_2^r into W_2^c,
CAL4: $W^c = W_1^c + W_2^c$,

where the superscripts c and r denote columnwise and rowwise partitioning, respectively. The above operation procedure twice includes the matrix transpose which normally requires all-to-all data communication. In the MPI standards, the all-to-all data communication is realized by a collective communication `MPI_Alltoallv`. However, due to irregular and incontiguous structure of the transferring data, the data-transpose communication should be executed by a point-to-point or a one-side communication function. On the Earth Simulator, since the one-side communication function `MPI_Put` more excellently runs than the point-to-point communication, `MPI_Put` is recommended by the developers [21]. Thus, we used `MPI_Put` .

In the procedure of the matrix-matrix multiplication, the calculation stages **CAL1** and **CAL2** and the communication one **COM1** is clearly found to be independently executed. Moreover, although the relation between **CAL3** and **COM2** is not so simple, the overlap can be realized in a pipelining fashion as shown in Fig. 3. Thus, the two communication processes (**COM1** and **COM2**) can be principally hidden behind the calculations.

3.3 A Technique to Save Memory

The `MPI_Put` function installed on the Earth Simulator can work as the blocking communication as well as the non-blocking one[2]. When using `MPI_Put` in the non-blocking mode, the call of `MPI_Win_Fence` to synchronize all processes is required in each pipeline stage. Otherwise, two N-word communication buffers (send- and receive-buffers) should be retained until the

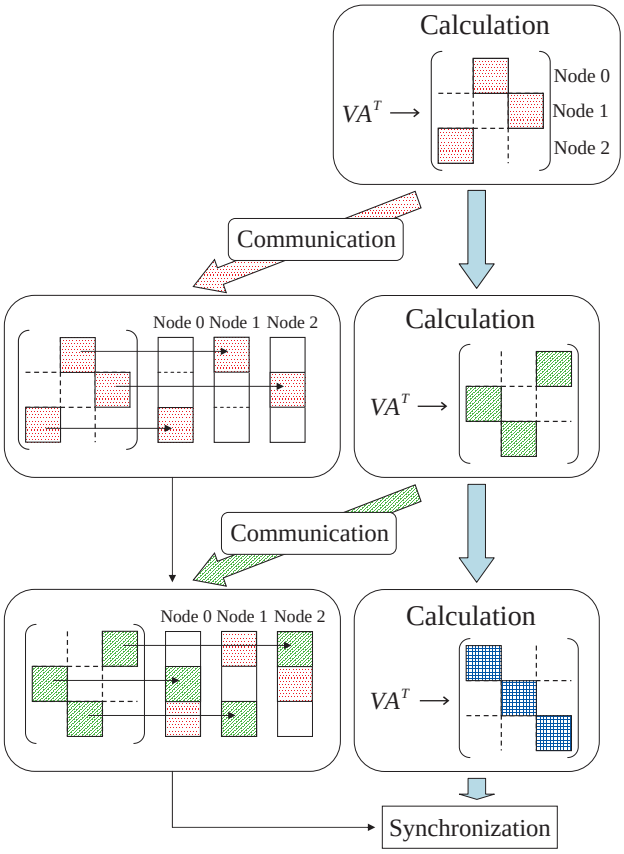

Fig. 3 A data-transfer diagram to overlap VA^T (**CAL3**) and communication (**COM2**) in a case using three nodes.

completion of all the stages. On the other hand, the completion of each stage is assured by return of the `MPI_Put` in the blocking mode, and send-buffer can be repeatedly re-used. Consequently, one N-word communication buffer (the send-buffer W_2^r) becomes free in the blocking mode. Thus, we can adopt the blocking `MPI_Put` to extend the maximum limit of the accessible matrix size. At a glance, this choice seems to sacrify the overlap functionality of the MPI library. However, one can manage to overlap computation with communication even in the use of the blocking `MPI_Put` on the Earth Simulator. The way is as follows. The blocking `MPI_Put` can be assigned to a single PE per node by the intra-node parallelization technique. Then, the assigned processor dedicates only the communication (see Fig. 4). As a result, the calculation load is divided into seven PE's. The intra-node parallelism is described by the automatic parallelization using CDIR compiler directives, and the eight tasks run in parallel. This parallelization method, which we call task assignment (TA) method, imitates a non-blocking communication operation, and enables to overlap between the blocking communication

[2] The blocking communication can be realized by a compile option '-noasync'.

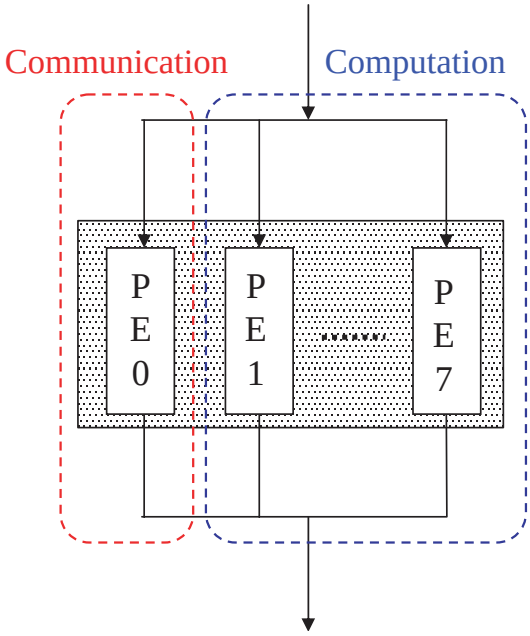

Fig. 4 A schematic figure for the task division. The communication task is assigned to a processor element (ex. PE 0) on each node.

and the calculation on the Earth Simulator (see Appendix C for more details). Here, we note that although the TA method just loses the floating-operation potential of a single processor, the total elapsed time does not almost change compared to the case using the non-blocking communication if the communication time is larger than 8/7 times of the floating operation time. Fortunately, our program just satisfies such a condition, i.e., the communication significantly occupies the total elapsed time. A performance comparison between the blocking `MPI_Put` with the TA method and the non-blocking `MPI_Put` is shown in Table 2. This result clearly shows that the elapsed times of both methods are almost comparable as we expect.

Table 2 A comparison of total elapsed time between the blocking `MPI_Put` with the TA method and the non-blocking `MPI_Put` . The calculation problem is the same as Model 1 in Table 3.

Method	Elapsed Time (sec)	
	Blocking+TA	Non-blocking
Lanczos	106.905	108.121
CG	39.325	39.250

3.4 Effective Usage of Vector Pipelines

The theoretical Flops rate in the single processor of the Earth Simulator is calculated by

$$F = \frac{4\,(\mathrm{ADD} + \mathrm{MUL})}{\max\,(\mathrm{ADD}, \mathrm{MUL}, \mathrm{VLD} + \mathrm{VST})}\ \ \mathrm{GFlops}, \qquad (4)$$

where ADD, MUL, VLD, VST denote the number of additions, multiplications, and vector load and store operations, respectively. According to the formula (4), the performance of the matrix multiplications AV and $V\,A^T$ shown in Fig. 5 is normally 2.67 GFlops. However, DO-loop unrolling decreases the number of VLD and VST

```
do i=1,n
  do j=ir(i),ir(i+1)-1 !Unroll
    do k=1,n
      w(i,k)=w(i,k)+v(ic(j),k)*a(j)
    enddo
  enddo
enddo
```

(a) $W = AV$

```
do i=1,n
  do j=ir(i),ir(i+1)-1 !Unroll
    do k=1,n
      w(k,i)=w(k,i)+v(k,ic(j))*a(j)
    enddo
  enddo
enddo
```

(b) $W = VA^T$

- Unrolling factor of 1 (non-unrolling)
  VLD=2, VST=1, MUL=1, ADD=1

$$F = \frac{4(1+1)}{\max(1,\,1,\,2+1)} = 2.6666\cdots$$

- Unrolling factor of 12
  VLD=13, VST=1, MUL=12, ADD=12

$$F = \frac{4(12+12)}{\max(12,\,12,\,13+1)} = 6.8571\cdots$$

(c) Flops Estimation

Fig. 5 (a), (b) Programs and (c) Flops estimation for the multiplications AV and $V\,A^T$. Here, n is the dimension size of the matrix A. See text for VLD, VST, MUL, and ADD.

instructions and improves the performance. In fact, when the loop unrolling factor is 12 in the multiplication, the performance is estimated to be 6.86 GFlops (see Fig. 5 (c)). Moreover, the loop fusion and re-construction of the algorithms improve the performance further.

4. Performance on the Earth Simulator

Let us present the performance of the Lanczos method and the PCG method with the TA method for huge Hamiltonian matrices (see Table 3 for the problem size and the matrix dimension). Table 4 shows the performance of these methods on 128 nodes, 256 nodes, and 512 nodes of the Earth Simulator. The performance measurements are made as follows. The total elapsed time and Flops rates are measured by using the performance analysis routines [20] installed on the Earth Simulator. On the other hand, the elapsed time of the solvers are measured by using `MPI_Wtime` function, and the Flops rates of the solvers are evaluated by the elapsed time and flop count estimated according to the following formulae:

- $5 * ndim + 16 * itr * ndim + 2 * itr * (2 * nnz - ndim)$ for the Lanczos method,
- $35 * ndim + 46 * itr * ndim + (itr + 2) * (2 * nnz - ndim)$ for the PCG method,

where $ndim$, itr and nnz are the dimension of the Hamiltonian matrix H, the number of iterations, and the number of the non-zero elements of H, respectively.

As shown in Table 4, the PCG method shows better convergence property and solves the eigenvalue problems about 3 times faster than the Lanczos method. Moreover, the PCG method overlaps communication tasks with calculation ones more than the Lanczos method since the communication can be efficiently hidden in a routine calculating inner products intrinsic to the PCG method. The best performance of the PCG method is 16.447 TFlops on 512 nodes which is 50.2% of the theoretical peak. On the other hand, Table 3 and 4 show that the Lanczos method can solve up to the 159-billion-dimensional Hamiltonian matrix on 512 nodes. To our knowledge, this size is the largest in a history of the exact diagonalization method of Hamiltonian matrices.

5. Numerical Results

In this section, we numerically study the repulsive

Table 3 The dimension of Hamiltonian matrix H, the number of nodes, and memory requirements from Model 1 to 4. In the PCG method, Model 4 requires 10.7 TB, which is beyond the memory size of 512 nodes of the Earth Simulator.

Model	No. of Sites	No. of Fermions		Dimension of H	No. of Nodes	Memory (TB)	
		$\uparrow$-spin	$\downarrow$-spin			Lanczos	PCG
1	24	6	6	18,116,083,216	128	0.8	1.3
2	21	8	8	41,408,180,100	256	1.9	2.9
3	22	8	8	102,252,852,900	512	4.6	6.9
4	22	9	8	159,059,993,400	512	7.1	(10.7)

Table 4 Performances of the Lanczos method and the PCG method with the TA method on the Earth Simulator.

a) The number of iterations, residual error, and elapsed time.

Model	Lanczos Method				PCG Method			
	Itr.	Residual Error	Elapsed Time(sec)		itr.	Residual Error	Elapsed Time(sec)	
			Total	Solver			Total	Solver
1	200	1.1×10^{-7}	106.905	101.666	105	1.4×10^{-9}	39.325	34.285
2	200	7.7×10^{-7}	154.159	148.453	107	2.3×10^{-9}	55.888	48.669
3	300	3.6×10^{-11}	288.270	279.775	109	2.4×10^{-9}	66.819	59.510
4	300	4.2×10^{-8}	362.635	352.944	——			

itr. ⋯ the number of iterations

b) Flops rate.

Model	TFlops (Peak Ratio)			
	Lanczos Method		PCG Method	
	Total	Solver	Total	Solver
1	3.062(37.4%)	3.208(39.2%)	4.045(49.4%)	4.607(56.2%)
2	5.245(32.0%)	5.426(33.1%)	6.928(42.3%)	7.893(48.2%)
3	10.613(32.3%)	10.906(33.3%)	14.763(45.1%)	16.447(50.2%)
4	13.363(40.8%)	13.694(41.8%)	–	–

Hubbard model with the trapping potential [6, 9] to examine a possibility of superfluidity in an atomic Fermi gas loaded on an optical lattice [8] and a nano-scale strongly-correlated electronic system. We diagonalize the Hubbard Hamiltonian H including the trapping potential (1) and calculate a binding energy, which is a probe for superfluidity, with varying the repulsive $U(> 0)$ and V. The probe is known to be not enough to confirm superfluidity of the system, but, the negative binding energy is an essential condition. We also measure the Cooper pair function in order to find a condition under which Cooper pair (superfluid) function develops. The result is directly correlated to the negative binding energy [6].

The optical lattice [8] is a standing wave formed by the interference effect of two laser beams. It gives a periodical lattice potential for atoms and enables to experimentally study the Hubbard model [7] showing strongly correlated behaviors. Since the discovery of high-T_c cuprates, the Hubbard model has attracted much attention as a simple model to investigate the mechanism of superconductivity originating from the Coulomb repulsion. However, it is still elusive whether or not the repulsive interaction truly leads to a pairing interaction which can bring about high-T_c superconductivity reaching $T_c \sim$ 150K. On the other hand, in the atomic Fermi gas [4, 5] loaded on the optical lattice, various physical parameters, such as the strength of the hopping, the interaction, and the particle density, are systematically tunable. Thus, the quest for the Cooper pairing via a repulsive interaction in the presence of the optical lattice will be a next big experimental challenge in the field of the atomic gas physics[25] , and the result will have a great impact on the solid state physics.

Since the atomic Fermi gas is trapped inside harmonic-well type of potential as schematically seen in Fig. 1, one should actually include the effect of the trap potential. In the presence of the trap, atoms tends to form a cluster in the center of the trap, which naturally leads to a *dome-like* density profile around the trap center [9] in relatively small U/t regime as seen in Fig. 6(a). On the other hand, since the presence of the on-site repulsion excludes the double occupancy of atoms, a *flat* density profile with one atom per one site consequently becomes favorable in a large on-site-repulsion regime as seen in Fig. 6(b). The flat region is called Mott core below. Thus, since the trap potential and the repulsive interaction have opposite effects as explained above, their interplay is an interesting problem. See Fig. 6(c) for the entire shape change of the particle density profile as a function of U/t. It is found that the spatial profile of the particle density abruptly changes from the dome like shape to the Mott core one with increasing U/t.

In this paper, we show a possibility of fermion superfluidity associated with the opposite effects of the U/t and

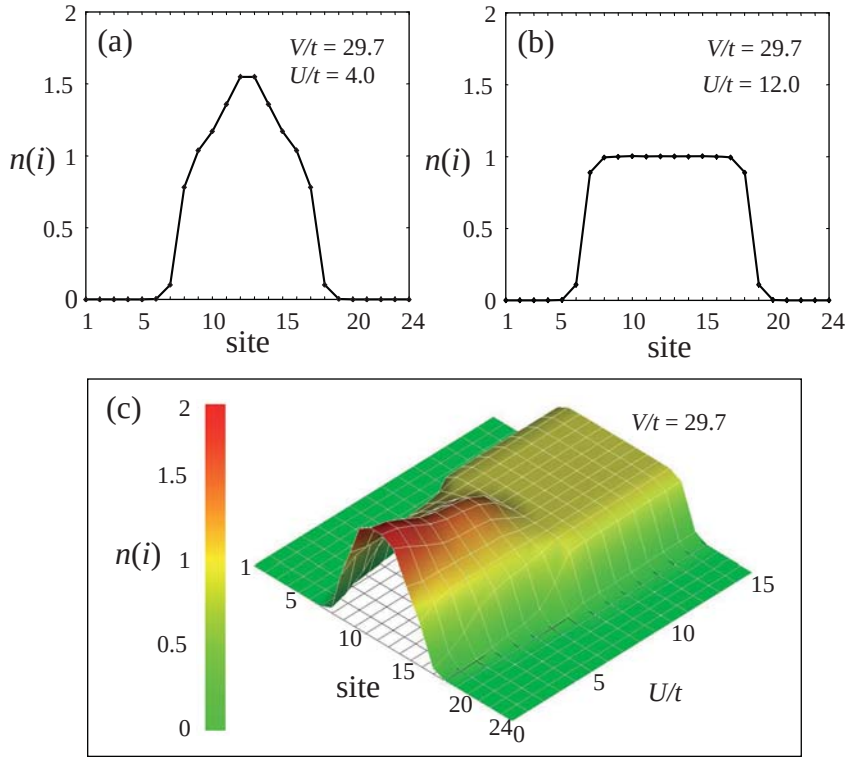

Fig. 6 Particle density profile (a) U/t = 12, (b) U/t = 4, and (c) $0 \le U/t \le 15$ for 12 fermions(6↑, 6↓) systems in 24-site Hubbard model with the trapped potential (V/t = 29.7).

V/t. We expect that a superfluid phase emerges when the Mott core is formed as seen in Fig. 6(b). In the same situation, the compressibility in the atom density is almost zero in the center of trap [9], whereas it shows a finite value around the edges of the core [9]. This makes it possible to fluctuate the atom number only around the Mott core edges and leads to a singlet Cooper pairing via the Mott core [6] at the core edges.

In order to confirm the above scenario about the superfluidity, we perform the exact diagonalization on the repulsive Hubbard model with the trapping potential. First, we fix the trapping strength, i.e., V/t = 29.7, which corresponds to the trap potential magnitude at the lattice edge, and examine how the binding energy of two Fermions E_b [10] changes with varying U/t. The binding energy of two Fermi atoms is given by

$$E_b \equiv E_g(n+1\uparrow, n+1\downarrow) + E_g(n\uparrow, n\downarrow) - 2E_g(n\uparrow, n+1\downarrow). \quad (5)$$

Here, $E_g(n\uparrow, n\downarrow)$ is the ground state energy in the case of N_F (= $n\uparrow$ +$n\downarrow$) fermions, which is evaluated by the exact diagonalization scheme using both the conventional Lanczos method [10] or the new algorithm (the PCG method). We mainly use the PCG method for systematic calculations because of its fast turn-round. If E_b is negative, it then means that an attractive interaction works between two atoms. Furthermore, it is expected for the negative E_b that the Cooper pair function develops.

In the numerical experiments, the number of site N is taken 24, the range of i is from 1 to 24, and the number of fermions N_F varies from 4(2↑, 2↓) to 14(7↑, 7↓). In the case of N_F = 14, we use the Lanczos method since the case is executable only by using the Lanczos method with the technique for the memory save described in Section 3.4. Fig. 7 shows U/t dependences of E_b from N_F = 4 to 12. Here, we note that E_b in N_F = 12 requires E_g in N_F = 14 as expressed in (5). It is found that E_b goes to negative in a large U/t region and the amplitude of the negative E_b slightly increases with increasing N_F up to N_F = 10. Here, we note that the negative E_b emerges above a certain V. If V is very small, then the system just behaves as the bulk one-dimensional system which shows not superfluidity but anomalous metallic features [9]. Thus, it is found that both the trapping potential and the repulsive force contribute to the negative E_b.

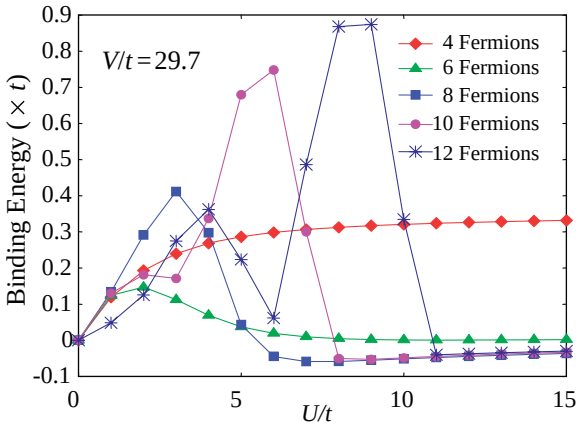

Fig. 7 U/t dependences of the binding energy E_b of two fermions for 4 fermions (2↑, 2↓), 6 fermions (3↑, 3↓), 8 fermions(4↑, 4↓), 10 fermions(5↑, 5↓), and 12 fermions(6↑, 6↓) in 24-site Hubbard model with the trapped potential (V/t = 29.7).

The next issue is whether the Cooper pair function develops or not. If it grows, then the interest is how the function develops. As shown in Fig. 8, the Cooper pair function grows between both edges of the Mott core

$$\Delta_{i,j} = \frac{1}{\sqrt{2}} \langle n+1\uparrow, n+1\downarrow \left| C^{\dagger}_{i\uparrow} C^{\dagger}_{j\downarrow} - C^{\dagger}_{i\downarrow} C^{\dagger}_{j\uparrow} \right| n\uparrow, n\downarrow \rangle.$$

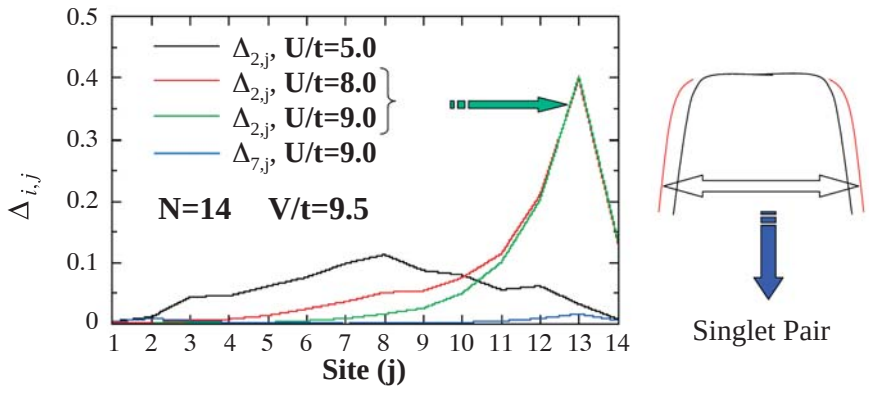

Fig. 8 U/t dependence of the site-site function for the singlet Cooper pair. When the Mott core appears, the function grows as schematically shown in the right hand panel.

when the E_b becomes negative with increasing U/t. It is found that the development of the Cooper pair function is strongly correlated to the appearance of the Mott core and the negative E_b. This result indicates that a higher dimensional system, e.g., the one as schematically shown in Fig. 9, may show superfluidity at the edges around the Mott core. We note that although the growth of the Cooper pair function and the negative binding energy E_b is not a direct evidence of the superfluidity, the superfluid instability occurs at least.

Now, let us study why the binding energy E_b becomes negative, i.e., why the attractive interaction works when the Mott core emerges. Fig. 10 shows the nearest neighbor correlation of the spin density given by

$$< s_i \cdot s_{i+1} >, \quad (6)$$

where $s_i = a^{\dagger}_{i,\uparrow} a_{i,\uparrow} - a^{\dagger}_{i,\downarrow} a_{i,\downarrow}$ [26]. We find as shown in Fig. 10 that the neighboring spin correlation shows a zigzag structure [26] when the Mott core develops. By noticing that Eq. (6) gives –3/4 for the spin singlet, it is found that the spin singlet are partially formed as the schematic figure. Next, let us compare the spin structure between $N(n, n)$ and $N + 2(n + 1, n + 1)$. From the schematic figures for N and N +2, it is found that the reformation of spin singlet pairs occurs by adding two fermions [26]. Through the reformation, the system

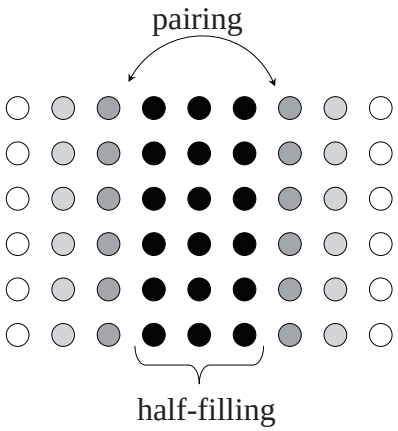

Fig. 9 A Schematic sketch for the pairing expected in higher (2D) dimensional lattice system.

obtains an energy gain because the formation of the spin singlet decreases the energy of the system [26]. Thus, the negative binding energy, i.e., the attractive interaction working between two fermions is found to be mainly originated from the reformation.

Finally, let us discuss a significance of the numerical results. The amplitude value of negative E_b depends on the trapping potential. The maximum amplitude is found to be $|E_b| = \sim 0.05t$ at $U/t \sim 8$ as seen in Fig. 7. Depending on the trapping potential, this value may exceed the binding energy scale in High-T_c cuprate superconductors, i.e.,

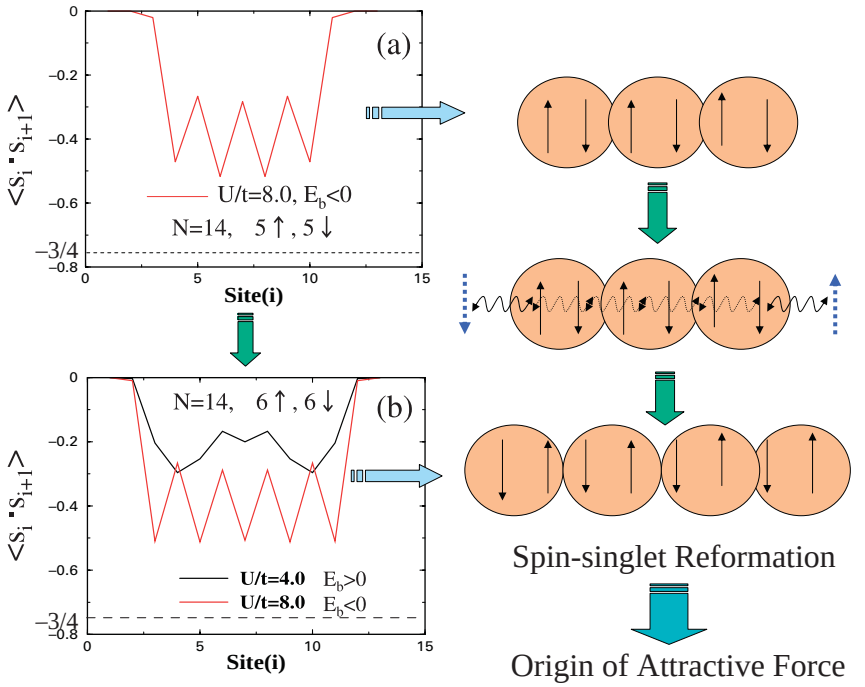

Fig. 10 The site profile of the nearest-neighbor spin correlation. (a) N = 14, (5↑, 5↓), U/t = 8.0 (E_b < 0) (b) N = 14, (6↑, 6↓) and the two cases for (E_b > 0) and (E_b < 0) are shown. The right hand side panel indicates how the dimerized spin structures are reformed when two particles are added.

the superconductivity with T_c ~ 150K. This result clearly indicates that it is possible to lift up or down the superfluid transition temperature by controlling the confinement effect. These surprising results can be easily confirmed by loading the atomic Fermi gas on the optical lattice.

6. Conclusions

We have investigated a pairing mechanism in the repulsive Hubbard model with confinement potential, which is a model for strongly correlated electron systems confined inside a nano-scale domain and atomic Fermi gases loaded on an optical lattice, by using exact diagonalization method. We have examined the preconditioned conjugate gradient method as an alternative scheme for the conventional Lanczos one and found that the total CPU time can be dramatically reduced compared to the Lanczos by choosing a suitable preconditioner. Furthermore, we have found the technique combining inter- and intra-node parallelisms to expand the maximum limit of solvable matrices. By developing these high-performance computing techniques, we have obtained the excellent performance in the new algorithm (beyond 16 TFlops) and the world record of the large matrix operation (beyond 1000 billion-dimensional). The calculation results have revealed that when the Coulomb repulsion exceeds a critical value and the confined potential is strong enough to make a Mott insulator region around the potential center, a Cooper pairing function develops between the both sides of the Mott region. This pairing is found to be strongly correlated to the spin structure inside the Mott region, which shows a dimerized character of spin-singlet. We believe that the result is a new and attractive insight in the condensed matter physics.

Acknowledgements

The authors in CCSE JAEA thank G. Yagawa, T. Hirayama, N. Nakajima, and C. Arakawa for their supports and acknowledge K. Itakura and all staff members in the Earth Simulator for their supports in the present calculations. One of the authors (M. M.) acknowledges T. Egami and P. Piekarz for illuminating discussion about diagonalization for d-p model, and T. Ishida and Y. Morita for their financial support. Two of authors (S. Y. and M. M.) thank Y. Oyanagi and T. Hotta for their illuminating discussion in the diagonalization technique.

(This article is reviewed by Dr. Tetsuya Sato.)

References

[1] M. H. Anderson, J. R. Ensher, M. R. Matthews, C. E. Wieman, and E. A. Cornell, Observation of Bose-Einstein Condensation in a Dilute Atomic Vapor, *Science*, **269**, 198–201, 1995.

[2] C. C. Bradley, C. A. Sackett, J. J. Tollett, and R. G. Hulet, Evidence of Bose-Einstein condensation in an atomic gas with attractive interactions, *Phys. Rev. Lett.*, **75**, 1687–1690, 1995.

[3] K. B. Davis, M. -O. Mewes, M. R. Andrews, N. J. van Druten, D. S. Durfee, D. M. Kurn, and W. Ketterle, Bose-Einstein condensation in a gas of sodium atoms, *Phys. Rev. Lett.*, **75**, 3969–3973, 1995.

[4] C. A. Regal, M. Greiner, and D. S. Jin, Observation of resonance condensation of fermionic atom pairs, *Phys. Rev. Lett.*, **92**, 040403, 2004.

[5] J. Kinast, S. L. Hemmer, M. E. Gehm, A. Turlapov, and J. E. Thomas, Evidence for superfluidity in a resonantly interacting Fermi gas, *Phys. Rev. Lett.*, **92**, 150402, 2004.

[6] 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.

[7] See, for example, M. Rasetti, ed., *The Hubbard Model: Recent Results*, World Scientific, Singapore, 1991; A. Montorsi, ed., *The Hubbard Model*, World Scientific, Singapore, 1992.

[8] M. Greiner, O. Mandel, T. Esslinger, T. W. Hansch, and I. Bloch, Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms, *Nature*, 415, 39, 2002.

[9] M. Rigol, A. Muramatsu, G. G. Batrouni, and R. T. Scalettar, Local quantum criticality in confined fermions on optical lattices, *Phys. Rev. Lett.*, **91**, 130403, 2003.

[10] For example, see, E. Dagotto, Correlated electrons in high-temperature superconductors, *Rev. Mod. Phys.*, **66**, 763, 1994.

[11] The Earth Simulator Center Webpage, http://www.es.jamstec.go.jp/index.en.html

[12] S. Shingu, H. Takahara, H. Fuchigami, M. Yamada, Y. Tsuda, W. Ohfuchi, Y. Sasaki, K. Kobayashi, T. Hagiwara, S. Habata, M. Yokokawa, H. Itoh, and K. Otsuka, A 26.58 Tflops Global Atmospheric Simulation with the Spectral Transform Method on the Earth Simulator, *Proc. of SC2002*, 2002. http://sc-2002.org/paperpdfs/pap.pap331.pdf

[13] H. Sakagami, H. Murai, Y. Seo, and M. Yokokawa, 14.9 TFLOPS Three-dimensional Fluid Simulation for Fusion Science with HPF on the Earth Simulator, *Proc. of SC2002*, 2002. http://sc-2002.org/paperpdfs/pap.pap147.pdf

[14] M. Yokokawa, K. Itakura, A. Uno, T. Ishihara, and Y. Kaneda, 16.4 Tflops Direct Numerical Simulation of Turbulence by Fourier Spectral Method on the Earth Simulator, *Proc. of SC2002*, 2002. http://sc-2002.org/paperpdfs/pap.pap273.pdf

[15] D. Komatitsch, S. Tsuboi, C, Ji, and J. Tromp, A 14.6 billion degrees of freedom, 5 teraflops, 2.5 terabyte earthquake

simulation on the Earth Simulator, *Proc. of SC2003*, 2003. http://www.sc-conference.org/sc2003/paperpdfs/pap124.pdf

[16] A. Kageyama, M. Kameyama, S. Fujihara, M. Yoshida, M. Hyodo, and Y. Tsuda, A 15.2 TFlops Simulation of Geodynamo on the Earth Simulator, *Proc. of SC2004*, 2004. http://www.sc-conference.org/sc2004/schedule/pdfs/pap234.pdf

[17] Gordon Bell Prizes of Supercomputing 2006. http://sc06.supercomputing.org/conference/gordon_bell_prize.php

[18] 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 SC2006*, 2006. http://sc06.supercomputing.org/schedule/pdf/gb113.pdf.

[19] J. K. Cullum, and R. A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol.1*: *Theory*, SIAM, Philadelphia, 2002.

[20] NEC Corporation, *FORTRAN90/ES Programmer's Guide*, *EARTH SIMULATOR User's Manuals*, NEC Corporation, 2002.

[21] H. Uehara, M. Tamura, and M. Yokokawa, MPI Performance Measurement on the Earth Simulator, *NEC Research & Development*, **44**(1), 75–79, 2003.

[22] A. V. Knyazev, Preconditioned eigensolvers - An oxymoron?, *Electronic Transactions on Numerical analysis*, Vol.**7**, 104–123, 1998.

[23] A. V. Knyazev, Toward the optimal eigensolver: Locally optimal block preconditioned conjugate gradient method, *SIAM J. Sci. Comput.*, **23**, 517–541, 2001.

[24] S. Yamada, T. Imamura, and M. Machida, Preconditioned Conjugate Gradient Method for Largescale Eigenvalue Problem of Quantum Problem: Convergence Property of Adaptive-shift Preconditioner (in Japanese), *Transactions of JSCES*, 20060027, 2006.

[25] W. Hofstetter, J. I. Cirac, P. Zoller, E. Demler, and M. D. Lukin, High-temperature superfluidity of fermionic atoms in optical lattices *Phys. Rev. Lett.*, **89**, 220407, 2002.

[26] M. Machida, S. Yamada, Y. Ohashi, and H. Matsumoto, Machida et al. Reply, *Phys. Rev. Lett.*, **95**, 218902, 2005.

Appendix A: How to Make a Matrix

Let us briefly show how to make a matrix in the exact diagonalization method. The state depicted in Fig. 1 is described as $A \equiv |001011>_{\uparrow} |111000>_{\downarrow}$. All possible configurations are taken into consideration, and the matrix elements are created as the expectation values of the Hamiltonian, e.g., $<A|H|A>$ is a diagonal component, and $<A|H|B>$, where $|B>$ is a different state, is a non-diagonal one. Here, we note that since the Hamiltonian (1) breaks translational symmetry due to the presence of the trap potential, we can not do any matrix size reductions using the symmetry. In terms of the fermion configuration, there are following rules. The numbers of fermions and sites are fixed. Two fermions with the same pseudo-spin can not occupy a site. This means that the occupation by fermions with different pseudo-spins on a site becomes the maximum occupation per site. The Pauli principle gives these rules.

Appendix B: Choice of Preconditioner

It is well-known that the preconditioning improves convergence of the CG method. However, it is generally hard to predict an effective preconditioning before actual test calculation. In this section, we focus on the following five preconditioners:

1. Point Jacobi,
2. Zero-shift point Jacobi,
3. Block Jacobi,
4. Neumann-polynomial expansion,
5. SSOR-type iteration.

Here, 1, 3, 4, and 5 are very popular preconditioners for the CG method and 2 (zero-shift point Jacobi) is a modified version of 1 (point Jacobi). The zero-shift point Jacobi is a diagonal scaling preconditioner shifted by '$-\mu_k$' to amplify the ground-state eigenvector, i.e., the preconditioning matrix is given by $T = (D-\mu_k I)^{-1}$, where D, I, and μ_k are the diagonal part of the matrix H, the identity matrix, and an approximate of the smallest eigenvalue which appears in the PCG iteration, respectively. In order to solve a huge matrix and achieve higher performance, we select 1 and 2, since they do not require any data communication and any extra storage.

Now, let us show a result of the preconditioner test, in which we solve the same eigenvalue problem as Section 2.3 to compare 1 and 2. Table 5 summarizes a performance test of three cases, without preconditioner (NP), point Jacobi (PJ), and zero-shift point Jacobi (ZS-PJ) on the Earth Simulator, and Fig. 11 illustrates these convergence properties. These results clearly reveal that the zero-shift point Jacobi is the best preconditioner.

Appendix C: Overlap between Blocking Communication and Calculation

In order to examine the possibility of overlap between blocking communication and calculation when utilizing TA method, we perform the multiplication Hv on 10 nodes (80 PE's) of the Earth Simulator. A test matrix H is a 1.5-billion-dimensional matrix derived from the one-dimensional 20-site Hubbard model with 12 fermions (6 $\uparrow$, 6 $\downarrow$). We measure the elapsed time of the four calculation steps **CAL1-4** and the two communications **COM1-2** shown in Section 3.2. We show the timecharts of TA

and non-assignment (NA) methods using the blocking communication in Fig. 12. As shown in the figure, the calculations and the communications are executed serially in NA method, while the calculation and the communication in TA method are executed simultaneously.

Table 5 Comparison among three preconditioners.

	NP	PJ	ZS-PJ
# Iteration	268	133	91
Residual Error	1.445E-09	1.404E-09	1.255E-09
Elapsed Time (sec)	78.904	40.785	28.205
FLOPS	382.55G	383.96G	391.37G

The test eigenvalue problem is the same as the one in Section 2.3.

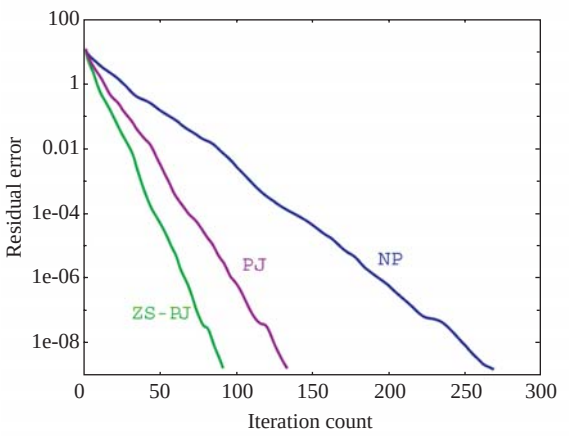

Fig. 11 Convergence properties of three preconditioners.

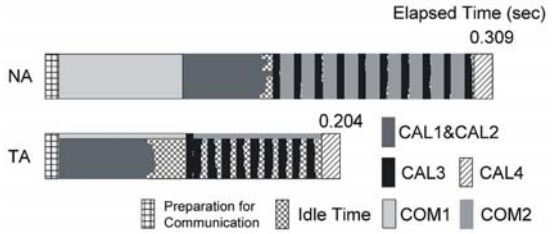

Fig. 12 A comparison of the process timechart on node No.0 for the multiplication Hv between the blocking communication with the TA method and non-assignment method (NA).