High Performance LOBPCG Method for Solving Multiple Eigenvalues of Hubbard Model: Efficiency of Communication Avoiding Neumann Expansion Preconditioner

Susumu Yamada^{1(✉)}, Toshiyuki Imamura², and Masahiko Machida¹

¹ Japan Atomic Energy Agency, Kashiwa, Chiba, Japan
`yamada.susumu@jaea.go.jp`

² RIKEN, Kobe, Hyogo, Japan

Abstract. The exact diagonalization method is a high accuracy numerical approach for solving the Hubbard model of a system of electrons with strong correlation. The method solves for the eigenvalues and eigenvectors of the Hamiltonian matrix derived from the Hubbard model. Since the Hamiltonian is a huge sparse symmetric matrix, it was expected that the LOBPCG method with an appropriate preconditioner could be used to solve the problem in a short time. This turned out to be the case as the LOBPCG method with a suitable preconditioner succeeded in solving the ground state (the smallest eigenvalue and its corresponding eigenvector) of the Hamiltonian. In order to solve for multiple eigenvalues of the Hamiltonian in a short time, we use a preconditioner based on the Neumann expansion which uses approximate eigenvalues and eigenvectors given by LOBPCG iteration. We apply a communication avoiding strategy, which was developed considering the physical properties of the Hubbard model, to the preconditioner. Our numerical experiments on two parallel computers show that the LOBPCG method coupled with the Neumann preconditioner and the communication avoiding strategy improves convergence and achieves excellent scalability when solving for multiple eigenvalues.

1 Introduction

Since the High- T_c superconductor was discovered many physicists have tried to understand the mechanism behind the superconductivity. It is believed that strong electron correlations underlie the phenomenon, however the exact mechanism is not yet fully understood. One of the numerical approaches to the problem is the exact diagonalization method. In this method the eigenvalue problem is solved for the Hamiltonian derived exactly from the Hubbard model [1,2], which is a model of a strongly-correlated electron system. When we solve the ground state (the smallest eigenvalue and its corresponding eigenvector) of the

```

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

```

Fig. 1. Algorithm of LOBPCG method for matrix A . Here the matrix T is a preconditioner.

Hamiltonian, we can understand its properties at absolute zero (-273.15°C). Many computational methods for the problem have been proposed. Since the Hamiltonian from the Hubbard model is a huge sparse symmetric matrix, an iteration method, such as the Lanczos method [3] or the LOBPCG method (see Fig. 1) [4, 5], is usually utilized for solving the eigenvalue problem.

The convergence of the LOBPCG method depends strongly on the use of a preconditioner. We previously confirmed that the zero-shift point Jacobi preconditioner, which is a shift-and-invert preconditioner [6] using an approximate eigenvalue, gives excellent convergence properties for the Hubbard model with the trapping potential [7]. However, we also reported that the benefit of the preconditioner strongly depends on the characteristics of the non-zero elements of the Hamiltonian and that the preconditioner does not always improve the convergence [8]. Therefore we proposed a novel preconditioner using the Neumann expansion for solving the ground state of the Hamiltonian and demonstrated that this preconditioner improves convergence for a Hamiltonian that is difficult to solve with the zero-shift point Jacobi preconditioner [8]. Moreover we applied a communication avoiding strategy, which was developed considering the properties of the Hubbard model, to the preconditioner.

In order to understand more details of strongly correlated electron systems in particular properties at temperatures near absolute zero, we must solve for the several smallest eigenvalues and corresponding eigenvectors of the Hamiltonian. The LOBPCG method can solve multiple eigenvalues by using a block of vectors.

In this paper, we extend the Neumann expansion preconditioner to the LOBPCG method for solving multiple eigenvalues and corresponding eigenvectors. Moreover, we demonstrate that the preconditioner improves the convergence properties and can achieve excellent parallel performance.

The paper is structured as follows. In Sect. 2 we briefly introduce related work for solving the ground state of the Hubbard model using the LOBPCG method. Section 3 describes the use of the Neumann expansion preconditioner with the communication avoiding strategy for solving for multiple eigenvalues and their corresponding eigenvectors. Section 4 demonstrates the parallel performance of the algorithm on the SGI ICE X and K supercomputers. A summary and conclusions are given in Sect. 5.

2 Related Work

2.1 Hamiltonian-Vector Multiplication

When solving the ground state of a symmetric matrix using the LOBPCG method, the most time-consuming operation is the matrix-vector multiplication. The Hamiltonian derived from the Hubbard model (see Fig. 2) is

$$H = -t \sum_{i,j,\sigma} c_{j\sigma}^\dagger c_{i\sigma} + \sum_i U_i n_{i\uparrow} n_{i\downarrow}, \quad (1)$$

where t is the hopping parameter from one site to another, and U_i is the repulsive energy for double occupation of the i -th site by two electrons [1, 2, 7]. Quantities $c_{i,\sigma}$, $c_{i,\sigma}^\dagger$ and $n_{i,\sigma}$ are the annihilation, creation, and number operator of an electron with pseudo-spin σ on the i -th site, respectively. The indices in formula (1) for the Hamiltonian denote the possible states for electrons in the model. The dimension of the Hamiltonian for the n_s -site Hubbard model is

$$\binom{n_s}{n_\uparrow} \times \binom{n_s}{n_\downarrow},$$

where $n_\uparrow$ and $n_\downarrow$ are the number of the up-spin and down-spin electrons, respectively.

The diagonal element in formula (1) is derived from the repulsive energy U_i in the corresponding state. The hopping parameter t affects non-zero elements with

Fig. 2. A schematic figure of the 2-dimensional Hubbard model, where t is the hopping parameter and U is the repulsive energy for double occupation of a site. Up arrows and down arrows correspond to up-spin and down-spin electrons, respectively.

column-index corresponding to the original state and row-index corresponding to the state after hopping. Since the ratio U/t greatly affects the properties of the model, we have to execute many simulations varying this ratio to reveal the properties of the model.

When considering the physical properties of the model, we can split the Hamiltonian-vector multiplication as

$$Hv = Dv + (I_{\downarrow} \otimes A_{\uparrow})v + (A_{\downarrow} \otimes I_{\uparrow})v, \quad (2)$$

where $I_{\uparrow(\downarrow)}$, $A_{\uparrow(\downarrow)}$ and D are the identity matrix, a sparse symmetric matrix derived from the hopping of an up-spin electron (a down-spin electron), and a diagonal matrix from the repulsive energy, respectively [7]. Since there is no regularity in the state change by electron hopping, the distribution of non-zero elements in matrix $A_{\uparrow(\downarrow)}$ is irregular.

Next, a matrix V is constructed by the following procedures from a vector v . First, decompose the vector v into n blocks, and order in the two-dimensional manner as follows,

$$v = \underbrace{(v_{1,1}, v_{2,1}, \dots, v_{m_{\uparrow},1})}_{\text{the first block}}, \underbrace{(v_{1,2}, v_{2,2}, \dots, v_{m_{\uparrow},2})}_{\text{the second block}}, \dots, \underbrace{(v_{1,m_{\downarrow}}, v_{2,m_{\downarrow}}, \dots, v_{m_{\uparrow},m_{\downarrow}})}_{\text{the } m_{\downarrow}\text{-th block}}^T,$$

where $m_{\uparrow}$ and $m_{\downarrow}$ are the dimensions of the Hamiltonian for up-spin and down-spin electrons, i.e.

$$m_{\uparrow} = \begin{pmatrix} n_s \\ n_{\uparrow} \end{pmatrix}, m_{\downarrow} = \begin{pmatrix} n_s \\ n_{\downarrow} \end{pmatrix}.$$

The subscripts on each element of v formally indicate the row and column within the matrix V . Therefore V is a dense matrix. The k -th elements of the matrix D , d_k , are used in the same manner to define a new matrix $\bar{D}$. The multiplication in Eq. (2) can then be written as

$$V_{i,j}^{new} = \bar{D}_{i,j} V_{i,j} + \sum_k A_{\uparrow i,k} V_{k,j} + \sum_k V_{i,k} A_{\downarrow j,k} \quad (3)$$

where the subscript i, j of the matrix is represented as the (i, j) -th element and V and $\bar{D}$. Accordingly we can parallelize the multiplication $Y = HV (\equiv Hv)$ as follows:

CAL 1: $Y^c = \bar{D}^c \odot V^c + A_{\uparrow} V^c$,

COM 1: all-to-all communication from V^c to V^r ,

CAL 2: $W^r = V^r A_{\downarrow}^T$,

COM 2: all-to-all communication from W^r to W^c ,

CAL 3: $Y^c = Y^c + W^c$.

where superscripts c and r denote column wise and row wise partitioning of the matrix data for the parallel calculation. The operator $\odot$ means an element wise multiplication. The parallelization strategy requires two all-to-all communication operations per multiplication.

2.2 Preconditioner of LOBPCG Method for Solving the Ground State of Hubbard Model

Zero-Shift Point Jacobi Preconditioner. A suitable preconditioner improves the convergence properties of the LOBPCG method. As a consequence many preconditioners have been proposed. Preconditioners for the Hamiltonian derived from the Hubbard model also have been proposed. For the Hubbard model, the zero-shift point Jacobi (ZSPJ) preconditioner, which is a shift-and-invert preconditioner using an approximate eigenvalue obtained during LOBPCG iteration, has excellent convergence properties for Hamiltonians where the diagonal elements predominate over the off-diagonal elements, i.e. cases where the repulsive energy U is large [7,8].

Neumann Expansion Preconditioner. For the Hubbard model with a small repulsive energy, a preconditioner using the Neumann expansion was previously proposed [8]. The expansion is

$$(I - M)^{-1} = I + M + M^2 + M^3 + \dots \quad (4)$$

The expansion converges when the operator norm of the matrix M is less than 1 ($\|M\|_{op} < 1$) [9]. Here the matrix M is

$$M = I - \frac{2}{\lambda_{\max} - \lambda_{\min}}(H - \lambda_{\min}I),$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the smallest and largest eigenvalues, respectively. When the exact eigenvalues are utilized for $\lambda_{\min}$ and $\lambda_{\max}$, $\|M\|_{op}$ is equal to 1. Since the LOBPCG method calculates an approximation of the smallest eigenvalue, we consider the residual error of this approximation and make a low estimate of $\lambda_{\min}$. The Gershgorin circle theorem is used to assign a $\lambda_{\max}$ that is an overestimate of the true value. The inequality $\|M\|_{op} < 1$ is hence obeyed and the expansion (4) can converge, i.e. the inverse matrix of $\frac{2}{\lambda_{\max} - \lambda_{\min}}(H - \lambda_{\min}I)$. The expansion is an effective preconditioner for the smallest eigenvalue $\lambda_{\min}$. In practice the Gershgorin circle theorem may give estimates for $\lambda_{\max}$ that are much too large. Multiplying by a damping factor α can help alleviate this inefficiency. We found 0.9 to be an appropriate α in numerical tests.

2.3 Communication Avoiding Neumann Expansion Preconditioner for Hubbard Model

When we execute the LOBPCG method with the s -th order Neumann expansion preconditioner, we calculate $s + 1$ matrix-vector multiplications, Hv , H^2v , $\dots$, and $H^{s+1}v$, per iteration. As the multiplications $(I_{\downarrow} \otimes A_{\uparrow})$ and $(A_{\downarrow} \otimes I_{\uparrow})$ are commutative Yamada et al. proposed a communication avoiding strategy for the Hamiltonian-vector multiplication [8]. Then, H^2 is given as

$$\begin{aligned} H^2 &= (I_{\downarrow} \otimes A_{\uparrow})(D + (I_{\downarrow} \otimes A_{\uparrow})) + (A_{\downarrow} \otimes I_{\uparrow})(D + (A_{\downarrow} \otimes I_{\uparrow})) \\ &\quad + D(D + (I_{\downarrow} \otimes A_{\uparrow}) + (A_{\downarrow} \otimes I_{\uparrow})) + (I_{\downarrow} \otimes A_{\uparrow})(A_{\downarrow} \otimes I_{\uparrow}) + (A_{\downarrow} \otimes I_{\uparrow})(I_{\downarrow} \otimes A_{\uparrow}) \\ &= (I_{\downarrow} \otimes A_{\uparrow})(D + (I_{\downarrow} \otimes A_{\uparrow}) + 2(A_{\downarrow} \otimes I_{\uparrow})) \\ &\quad + (A_{\downarrow} \otimes I_{\uparrow})(D + (A_{\downarrow} \otimes I_{\uparrow})) + D(D + (I_{\downarrow} \otimes A_{\uparrow}) + (A_{\downarrow} \otimes I_{\uparrow})). \end{aligned}$$

As a result, $Y_1 = Hv$ and $Y_2 = H^2v$ can be calculated by the following algorithm:

CAL 1: $Y^c = \bar{D}^c \odot V^c + A_{\uparrow} V^c$,
COM 1: all-to-all communication from V^c to V^r ,
CAL 2: $W^r = V^r A_{\downarrow}^T$,
COM 2: all-to-all communication from W^r to W^c ,
CAL 3: $Y_1^c = Y^c + W^c$,
CAL 4: $Y^c = Y_1^c + W^c$,
CAL 5: $Y^c = \bar{D}^c \odot Y_1^c + A_{\uparrow} Y^c$,
CAL 6: $W^r = \bar{D}^r \odot V^r + W^r$,
CAL 7: $W^r = W^r A_{\downarrow}^T$,
COM 3: all-to-all communication from W^r to W^c ,
CAL 8: $Y_2^c = Y^c + W^c$.

The algorithm requires three all-to-all communication operations. On the other hand, when using the original algorithm described in Sect. 2.1, four all-to-all communication operations are required to calculate the same multiplication. However, the new algorithm has extra calculations, CAL 4 and CAL 6, as compared to the original one. Therefore when the cost of one all-to-all communication operation is larger than that of the extra calculations, we expect to achieve speedup with the communication avoiding strategy. The algorithm can not be directly applied to the multiplication H^{s+1} for $s \geq 2$. In this case, the multiplication H^{s+1} is calculated by appropriately combining Hv and H^2v operations.

3 Neumann Expansion Preconditioner for Multiple Eigenvalues of Hubbard Model

3.1 How to Calculate Multiple Eigenvalues Using LOBPCG Method

The LOBPCG method for solving the m smallest eigenvalues and corresponding eigenvectors carries out recurrence with m vectors simultaneously (see Fig. 3). In this algorithm, the generalized eigenvalue problem has to be solved. We can solve the problem using the LAPACK function `dsyev`, if the matrix S_B is a positive definite matrix. Although theoretically S_B is always a positive definite matrix, numerically this is not always the case. The reason is that the norms of the vectors $w_k^{(i)}$ and $p_k^{(i)}$ ($i = 1, 2, \dots, m$) become small as the LOBPCG iteration converges, and it is possible that trailing digits are lost in the calculation of S_B . Therefore we set the matrix S_B to the identity matrix by orthogonalizing the vectors per iteration. In the following numerical tests, we utilize the TSQR method for the orthogonalization [10, 11].

```

 $\mathbf{x}_0^{(i)} := \text{an initial guess}, \mathbf{p}_0^{(i)} := 0, i = 1, \dots, m$ 
 $\mathbf{x}_0^{(i)} = \mathbf{x}_0^{(i)} / \|\mathbf{x}_0^{(i)}\|, i = 1, \dots, m$ 
 $\mathbf{X}_0^{(i)} := A\mathbf{x}_0^{(i)}, \mathbf{P}_0^{(i)} := 0, i = 1, \dots, m$ 
 $\mu_{-1}^{(i)} := (\mathbf{x}_0^{(i)}, \mathbf{X}_0^{(i)}), \mathbf{w}_0^{(i)} := \mathbf{X}_0^{(i)} - \mu_{-1}^{(i)}\mathbf{x}_0^{(i)}, i = 1, \dots, m$ 
do  $k=0, \dots$  until convergence
   $\mathbf{W}_k^{(i)} := A\mathbf{w}_k^{(i)}, i = 1, \dots, m$ 
   $S_A := \{\mathbf{w}_k^{(1)}, \dots, \mathbf{w}_k^{(m)}, \mathbf{x}_k^{(1)}, \dots, \mathbf{x}_k^{(m)}, \mathbf{p}_k^{(1)}, \dots, \mathbf{p}_k^{(m)}\}^T$ 
   $\{\mathbf{W}_k^{(1)}, \dots, \mathbf{W}_k^{(m)}, \mathbf{X}_k^{(1)}, \dots, \mathbf{X}_k^{(m)}, \mathbf{P}_k^{(1)}, \dots, \mathbf{P}_k^{(m)}\}$ 
   $S_B := \{\mathbf{w}_k^{(1)}, \dots, \mathbf{w}_k^{(m)}, \mathbf{x}_k^{(1)}, \dots, \mathbf{x}_k^{(m)}, \mathbf{p}_k^{(1)}, \dots, \mathbf{p}_k^{(m)}\}^T$ 
   $\{\mathbf{w}_k^{(1)}, \dots, \mathbf{w}_k^{(m)}, \mathbf{x}_k^{(1)}, \dots, \mathbf{x}_k^{(m)}, \mathbf{p}_k^{(1)}, \dots, \mathbf{p}_k^{(m)}\}$ 
  Solve the generalized eigenvalue problem  $S_A \mathbf{v} = \mu S_B \mathbf{v}$  to obtain the  $m$  smallest
  eigenvalues  $\mu^{(i)}$  and the corresponding eigenvectors  $\mathbf{v}^{(i)} = (\alpha^{(i)}, \beta^{(i)}, \gamma^{(i)})^T (i =$ 
   $1, \dots, m)$ ,
   $\mathbf{x}_{k+1}^{(i)} := \sum_{j=1}^m \{\alpha^{(j)} \mathbf{w}_k^{(j)} + \beta^{(j)} \mathbf{x}_k^{(j)} + \gamma^{(j)} \mathbf{p}_k^{(j)}\}, i = 1, \dots, m$ 
   $\mathbf{p}_{k+1}^{(i)} := \sum_{j=1}^m \{\alpha^{(j)} \mathbf{w}_k^{(j)} + \gamma^{(j)} \mathbf{p}_k^{(j)}\}, i = 1, \dots, m$ 
   $\mathbf{X}_{k+1}^{(i)} := \sum_{j=1}^m \{\alpha^{(j)} \mathbf{W}_k^{(j)} + \beta^{(j)} \mathbf{X}_k^{(j)} + \gamma^{(j)} \mathbf{P}_k^{(j)}\}, i = 1, \dots, m$ 
   $\mathbf{P}_{k+1}^{(i)} := \sum_{j=1}^m \{\alpha^{(j)} \mathbf{W}_k^{(j)} + \gamma^{(j)} \mathbf{P}_k^{(j)}\}, i = 1, \dots, m$ 
   $\mu_k^{(i)} := (\mathbf{x}_{k+1}^{(i)}, \mathbf{X}_{k+1}^{(i)}) / (\mathbf{x}_{k+1}^{(i)}, \mathbf{x}_{k+1}^{(i)}), i = 1, \dots, m$ 
   $\mathbf{w}_{k+1}^{(i)} := T^{(i)-1}(\mathbf{X}_{k+1}^{(i)} - \mu_k^{(i)} \mathbf{x}_{k+1}^{(i)}), i = 1, \dots, m$ 
enddo

```

Fig. 3. LOBPCG method for solving the m smallest eigenvalues and corresponding eigenvectors. $T^{(i)}$ is a preconditioner for the i -th smallest eigenvalues. This algorithm requires m matrix-vector multiplication operations and m preconditioned ones per iteration.

3.2 Neumann Expansion Preconditioner of LOBPCG Method for Solving Multiple Eigenvalues

When we calculate multiple eigenvalues (and corresponding eigenvectors) using the LOBPCG method, we can individually apply a preconditioning operation to each vector corresponding to the eigenvectors. We set the matrix M_i using the Neumann expansion preconditioner for the i -th smallest eigenvalue λ_i of the Hamiltonian as

$$M_i = I - \frac{2}{\lambda_{\max} - \lambda_i} (H - \lambda_i I).$$

Since we obtain approximate eigenvalues after each iteration of the LOBPCG method, we consider the residual errors of these approximations to define an appropriate λ_i . The matrix M_i has $(i-1)$ eigenvalues whose absolute values are greater than or equal to 1. In this case, the Neumann expansion using M_i can not converge. The eigenvectors corresponding to the eigenvalues agree with those corresponding to the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_{i-1}$ of the Hamiltonian, and then, they are calculated during the LOBPCG iteration simultaneously. Accordingly, we orthogonalize the vectors $\mathbf{x}_k^{(i)}, \mathbf{w}_k^{(i)}$, and $\mathbf{p}_k^{(i)}$ ($i = 1, 2, \dots, m$) in the order that takes away the components of the vectors $\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}, \dots, \mathbf{x}_k^{(i-1)}$ from $\mathbf{w}_k^{(i)}$ given

by the Neumann expansion preconditioner using M_i . That is, we orthogonal the vectors utilizing the algorithm including the following operation

$$w_k^{(j)} := w_k^{(j)} - \sum_{i=1}^{j-1} (w_k^{(j)}, x_k^{(i)}) x_k^{(i)}. \quad (5)$$

The formula (5) can approximately remove the components of the eigenvectors corresponding to the eigenvalues, whose absolute values are greater than or equal to 1, from the preconditioned vectors. Therefore we expect that the Neumann expansion using M_i becomes an appropriate preconditioner for solving for multiple eigenvalues.

4 Performance Result

4.1 Computational Performance and Convergence Property

We examined the computational performance and convergence properties of the LOBPCG method. We solved the 2-D 4×5 -site Hubbard model with 5 up-spin electrons and 5 down-spin ones. The dimension of the Hamiltonian derived from the model is about 240 million. The number of non-zero off-diagonal elements is about 1.6 billion. We solved for one, five and 10 eigenvalues (and corresponding eigenvectors) of the Hamiltonian on 768 cores (64 MPI processes $\times$ 12 OpenMP threads) of the SGI ICE X supercomputer (see Table 1) in Japan Atomic Energy Agency (JAEA). Table 2 shows the results for a weak interaction case ($U/t = 1$) and a strong one ($U/t = 10$). Table 3 shows the elapsed times of some representative operations.

The results for $U/t = 1$ indicate that point Jacobi (PJ) and zero-shift point Jacobi (ZSPJ) preconditioners hardly improve the convergence compared to without using a preconditioner at all. When we solve for many eigenvalues, the PJ and ZSPJ preconditioners have little effect on the speed of the calculation. On the other hand, the Neumann expansion preconditioner can decrease the number of iterations required for convergence. Moreover, the larger the Neumann expansion series s , the fewer iterations required. When we solve for only

Table 1. Details of SGI ICE X

Processor	Intel Xeon E5-2680v3 (2.5 GHz, 30 MB L2 cache)
FLOPS per processor	480 GFLOPS
Number of cores per CPU	12
Number of processors per node	2
Memory of node	64 GB
Memory bandwidth	68 GB/s
Network	Infini Band 6.8 GB/s
Compiler	Intel compiler

Table 2. Elapsed time and number of iterations for convergence of LOBPCG method using zero-shift point Jacobi (ZSPJ), Neumann expansion (NE), or communication avoiding Neumann expansion (CANE) preconditioner. Here, s is the number of the Neumann expansion series.

(a) One eigenvalue (The ground state)									
	Number of iterations (top) & Elapsed time (sec) (bottom)								
	No precon.	PJ	ZSPJ	NE			CANE		
				$s = 1$	$s = 2$	$s = 3$	$s = 1$	$s = 2$	$s = 3$
$U/t = 1$	133	133	132	69	59	46	69	59	46
	9.16	9.19	9.13	8.79	11.06	10.78	7.40	9.18	8.94
$U/t = 10$	184	132	124	95	81	65	94	81	64
	13.03	9.08	8.61	12.07	14.65	15.62	10.05	12.62	12.79
(b) 5 eigenvalues									
	Number of iterations (top) & Elapsed time (sec) (bottom)								
	No precon.	PJ	ZSPJ	NE			CANE		
				$s = 1$	$s = 2$	$s = 3$	$s = 1$	$s = 2$	$s = 3$
$U/t = 1$	199	190	168	81	77	59	86	72	54
	171.75	164.57	145.28	89.44	103.31	93.69	87.89	91.21	77.16
$U/t = 10$	293	217	240	159	156	108	155	142	105
	250.77	186.35	204.97	172.90	208.80	171.37	156.41	179.12	149.59
(c) 10 eigenvalues									
	Number of iterations (top) & Elapsed time (sec) (bottom)								
	No precon.	PJ	ZSPJ	NE			CANE		
				$s = 1$	$s = 2$	$s = 3$	$s = 1$	$s = 2$	$s = 3$
$U/t = 1$	551	777	624	319	257	184	340	302	198
	1221.55	1672.69	1369.91	911.56	897.79	759.40	863.57	936.35	680.48
$U/t = 10$	398	298	313	232	184	161	201	177	137
	996.19	740.98	763.42	720.22	704.14	705.03	579.98	607.46	515.60

the smallest eigenvalue, the total elapsed time increases as s increases. The reason is that the elapsed time of the Hamiltonian-vector multiplication operation is dominant over the whole calculation for solving the only smallest eigenvalue (see Table 3). When we solve multiple eigenvalues, the TSQR operation becomes dominant. Therefore when the series number s becomes large, it is possible to achieve speedup of the computation.

Next, we discuss the results for $U/t = 10$. The results indicate that the PJ preconditioner improves the convergence properties. On the other hand, ZSPJ for small m improves convergence, however, its convergence properties when solving for multiple eigenvalues are almost the same as those for the PJ preconditioner. When we solve for multiple eigenvalues using the Neumann expansion preconditioner, the solution is obtained faster than using the PJ or ZSPJ preconditioners. Moreover, as the Neumann expansion series s increases, the Neumann expansion

Table 3. Elapsed time for operations per iteration. This table shows the results using the zero-shift point Jacobi (ZSPJ), Neumann expansion (NE), and communication avoiding Neumann expansion (CANE). Here, the Neumann expansion series s is equal to 1. For $m = 1$, instead of executing TSQR, we calculate S_B , moreover, ZSPJ preconditioner is calculated together with x, p, X, P .

	Elapse time per iteration (sec)								
	$m = 1$			$m = 5$			$m = 10$		
	ZSPJ	NE	CANE	ZSPJ	NE	CANE	ZSPJ	NE	CANE
Hw (& H^2w)	0.061	0.117	0.100	0.276	0.545	0.448	0.568	1.088	0.909
TSQR	—	—	—	0.407	0.408	0.407	1.498	1.503	1.502
S_A (& S_B)	0.007	0.007	0.007	0.073	0.073	0.073	0.255	0.257	0.254
x, p, X, P	0.008	0.007	0.007	0.107	0.122	0.121	0.301	0.331	0.332
Preconditioner	—	0.003	0.003	0.018	0.015	0.014	0.035	0.030	0.028

Table 4. Speedup ratio for the elapsed time per iteration using the Neumann expansion preconditioner and communication avoiding strategy.

	Speedup ratio								
	$m = 1$			$m = 5$			$m = 10$		
	$s = 1$	$s = 2$	$s = 3$	$s = 1$	$s = 2$	$s = 3$	$s = 1$	$s = 2$	$s = 3$
$U/t = 1$	1.19	1.20	1.21	1.08	1.06	1.11	1.13	1.13	1.20
$U/t = 10$	1.19	1.16	1.20	1.08	1.06	1.11	1.08	1.12	1.16

preconditioner improves the convergence properties and the total elapsed time decreases, especially when m is large.

Finally, we talk about the effect of the communication avoiding strategy. Table 4 shows the speedup ratio for the elapsed time using the Neumann expansion preconditioner per iteration and the communication avoiding strategy. In all cases the communication avoiding strategy realizes speedup. When we solve for only the smallest eigenvalue (and its corresponding eigenvector), the speedup ratio is almost the same as that for the matrix-vector multiplication, because the multiplication cost is dominant. On the other hand, when we solve for multiple eigenvalues, the calculation cost except the multiplication becomes dominant. Therefore the speedup ratio is a little smaller than that for only the multiplication. Furthermore, when the Neumann expansion series s is equal to 3, we confirm that the ratio improves. In this case, since four multiplications (Hw , H^2w , H^3w and H^4w) are executed per iteration, the ratio of the multiplication cost increases. Moreover, we can execute four multiplication operations by two communication avoiding multiplications. Therefore, the ratio for $s = 3$ is better than that for $s = 1$.

4.2 Parallel Performance

In order to examine the parallel performance of the LOBPCG method using the Neumann expansion preconditioner, we solved for the 10 smallest eigenvalues and corresponding eigenvectors of the Hamiltonian derived from the 4×5 -site Hubbard model for $U/t = 1$ with 6 up-spin and 6 down-spin electrons. We used the LOBPCG method with ZSPJ, NE, and CANE preconditioners using hybrid parallelization on SGI ICEX in JAEA and the K computer in RIKEN (see Table 5). The results are shown in Table 6. The results indicate that all preconditioners achieve excellent parallel efficiency. The communication avoiding strategy on SGI ICEX decreases the elapsed time per iteration by about 15%. On the other hand, the communication avoiding strategy on the K computer did not realize speedup when using a small number of cores. The ratio of the network bandwidth to FLOPS per node of the K computer is larger than that of SGI ICEX, so it is possible that the cost of the extra calculations (CAL 4 & CAL 6) is larger than that of the all-to-all communication operation. However since the cost of the all-to-all communication operation increases as the number of the cores increases, the strategy realizes speedup on 4096 cores. Therefore, the strategy has a potential of speedup for parallel computing using a sufficiently large number of cores, even if the ratio of the network bandwidth to FLOPS is large.

Although the LOBPCG method using NE has four times more Hamiltonian-vector multiplications per iteration than the method with ZSPJ, the former takes about twice the elapsed time of the latter. The reason is that the calculation operations except the multiplication is dominant in this case. Therefore, we conclude that in order to solve for multiple eigenvalues of the Hamiltonian derived from the Hubbard model using the LOBPCG method in a short computation time, it is crucial to reduce the number of the iterations for the convergence even if the calculation cost of the preconditioner is large.

Table 5. Details of K computer

Processor	SPARCTM 64 VIIIfx (2 GHz, 6 MB L2 cache)
FLOPS per processor	128 GFLOPS
Number of cores per CPU	8
Number of processors per node	1
Memory of node	16 GB
Memory bandwidth	64 GB/s
Network	Torus network (Tofu) 5 GB/s
Compiler	Fujitsu compiler

Table 6. Parallel performance of LOBPCG method on SGI ICEX and K computer. This table shows the number of iterations, the total elapsed time, and the elapsed time per iteration of LOBPCG method using zero-shift point Jacobi (ZSPJ), Neumann expansion (NE), or communication avoiding Neumann expansion (CANE) preconditioner. Here, the Neumann expansion series s is 3.

(a) SGI ICEX			
	Number of iterations (top)		
	Elapsed time (sec) (middle)		
	Elapsed time per iteration (sec) (bottom)		
	ZSPJ	NE	CANE
64 MPI $\times$ 12 OpenMP	591	226	225
	9501.694	5886.533	4921.302
	16.077	26.047	21.872
128 MPI $\times$ 12 OpenMP	605	246	229
	4611.478	3662.846	2909.048
	7.622	14.890	12.703
256 MPI $\times$ 12 OpenMP	601	244	226
	2259.070	2043.231	1603.456
	3.759	8.374	7.095
(b) K computer			
	Number of iterations (top)		
	Elapsed time (sec) (middle)		
	Elapsed time per iteration (sec) (bottom)		
	ZSPJ	NE	CANE
128 MPI $\times$ 8 OpenMP	503	209	230
	5775.971	3752.884	4596.063
	11.483	17.956	19.983
256 MPI $\times$ 8 OpenMP	551	224	303
	3231.566	2085.268	2974.883
	5.865	9.309	9.818
512 MPI $\times$ 8 OpenMP	862	243	250
	2548.534	1327.093	1130.652
	2.957	5.461	4.523

5 Conclusions

In this paper we applied the Neumann expansion preconditioner to the LOBPCG method to solve for multiple eigenvalues and corresponding eigenvectors of the Hamiltonian derived from the Hubbard model. We examined the convergence properties and parallel performance of the algorithms. Since the norm of the matrix used in the Neumann expansion should be less than 1, we transform

it using approximate eigenvalues calculated by the LOBPCG iteration and the upper bounds of the eigenvalues by the Gershgorin circle theorem. Moreover, we orthogonalize the iteration vectors in the order that removes the components of the eigenvectors corresponding to the eigenvalues, whose absolute values are greater than or equal to 1, from the preconditioned vectors.

The Neumann expansion preconditioner with the communication avoiding strategy can achieve speedup even for problems which are hardly improved by the conventional preconditioners. Furthermore, a numerical experiment indicated that the LOBPCG method using this preconditioner has excellent parallel efficiency on thousands cores, and the communication avoiding strategy based on the property of the Hubbard model realizes speedup for parallel computers if a sufficiently large number of cores are used. Therefore, we confirm that the preconditioner based on the Neumann expansion is suitable for solving the eigenvalue problem derived from the Hubbard model using the LOBPCG method.

Acknowledgments. Computations in this study were performed on the SGI ICE X at the JAEA and the K computer at RIKEN Advanced Institute for Computational Science (project ID:ra000005). This research was partially supported by JSPS KAKENHI Grant Number 15K00178.

References

1. Rasetti, M. (ed.): The Hubbard Model: Recent Results. World Scientific, Singapore (1991)
2. Montorsi, A. (ed.): The Hubbard Model. World Scientific, Singapore (1992)
3. Cullum, J.K., Willoughby, R.A.: Lanczos Algorithms for Large Symmetric Eigenvalue Computations, vol. 1: Theory. SIAM, Philadelphia (2002)
4. Knyazev, A.V.: Preconditioned eigensolvers - an oxymoron? *Electron. Trans. Numer. Anal.* **7**, 104–123 (1998)
5. Knyazev, A.V.: Toward the optimal eigensolver: locally optimal block preconditioned conjugate gradient method. *SIAM J. Sci. Comput.* **23**, 517–541 (2001)
6. Saad, Y.: Numerical Methods for Large Eigenvalue Problems: Revised Edition. SIAM (2011)
7. Yamada, S., Imamura, T., Machida, M.: 16.447 TFlops and 159-Billion-dimensional exact-diagonalization for trapped Fermion-Hubbard Model on the Earth Simulator. In: Proceedings of SC 2005 (2005)
8. Yamada, S., Imamura, T., Machida, M.: Communication avoiding Neumann expansion preconditioner for LOBPCG method: convergence property of exact diagonalization method for Hubbard model. In: Proceedings of ParCo 2017 (2017, accepted)
9. Barrett, R., et al.: Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. SIAM, Philadelphia (1994)
10. Langou, J.: AllReduce algorithms: application to Householder QR factorization. In: Proceedings of the 2007 International Conference on Preconditioning Techniques for Large Sparse Matrix Problems in Scientific and Industrial Applications, pp. 103–106 (2007)
11. Demmel, J., Grigori, L., Hoemmen, M., Langou, J.: Communication-avoiding parallel and sequential QR factorizations, Technical report, Electrical Engineering and Computer Sciences, University of California Berkeley (2008)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.