A parallel eigensolver using contour integration for generalized eigenvalue problems in molecular simulation

[1]Tetsuya SAKURAI [1)], Hiroto TADANO [1)], Tsutomu IKEGAMI [2)], and Umpei NAGASHIMA [2)]

1) Department of Computer Science, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan
2) National Institute of Advanced Industrial Science and Technology, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan

Abstract

In this paper, we consider a parallel method for computing interior eigenvalues and corresponding eigenvectors of generalized eigenvalue problems which arise from molecular orbital computation of biochemistry applications. Matrices in such applications are sparse but often have relatively large number of nonzero elements, and we may require some eigenpairs in a specific part of the spectrum. We use a contour integration to construct a desired subspace. Some properties of the subspace obtained by numerical integration are discussed, and then a parallel implementation is presented. We report the numerical aspects and parallel performance of our method with matrices derived from molecular orbital computation.

Keywords: Interior eigenvalue problems, contour integration, Rayleigh-Ritz procedure, biochemistry application.

MSC(2000): 65F15, 65E05, 65Y05

1 Introduction

In this paper, we consider a parallel method for computing some eigenpairs $(\lambda, \boldsymbol{x})$ satisfying $A\boldsymbol{x} = \lambda B\boldsymbol{x}$ in a specific part of the spectrum, where $A, B \in \mathbb{R}^{n\times n}$ are symmetric and B is positive definite, which arise from molecular orbital computation. Molecular orbital computation has been performed to study a reaction mechanism of enzymes, an electronic structure of the photosynthetic systems, etc.

The fragment molecular orbital (FMO) method [4] enables one to calculate the total energy of molecule without performing the calculation of whole molecule. In [1], a full electron calculation for a large molecule was performed by using the FMO method on a massive cluster computer. The FMO-MO method [3, 12], which works as a post-process of the FMO method, gives a good approximation for the canonical MOs without SCF iterations. A huge Fock matrix is generated, and we need limited number of eigenpairs corresponding to frontier orbitals. Therefore a large-scale interior eigenvalue problem appears in the FMO-MO method.

[1]Corresponding Author Email: sakurai@cs.tsukuba.ac.jp

With growing parallel efficiency in the Fock matrix construction, diagonalization process becomes bottleneck in MO calculations. Since the Fock matrix has relatively large number of nonzero elements due to the base function of middle-range interaction of molecules, a sparse factorization of the shifted matrix for spectral transformation may not be feasible. An alternative to factorization is offered by iterative solvers. Unfortunately, it is often unattractive for parallel computation, because iterative process is performed successively. Therefore shift-and-invert approach is not effective to treat interior eigenvalue problems in such applications.

Our method for finding eigenpairs in a given physical domain is based on contour integral presented in [6]. One of major advantages of our eigensolver is that it does not require inner/outer loops; an inner loop to construct an approximate subspace, and an outer loop to update approximate eigenvectors. Moreover, the values which are used in a numerical integration can be evaluated independently on each integration node, it provides a variety of parallel programming models [7]. We recently proposed a Rayleigh-Ritz type method [8] in order to improve numerical stability. The block method [2] performs well if there are multiple eigenvalues in the interested spectrum.

The computation at each integral node involves linear system solutions where the coefficient matrices are derived from A and B. In [5], we found that a Krylov subspace iterative method in conjunction with a preconditioning using a complete factorization for an approximated coefficient matrix is effective to solve such linear systems.

In the next section, we explain our eigensolver using a contour integration. Then we present some numerical properties of the subspace obtained by a numerical approximation of the contour integration. In Section 3, we show an implementation of our method. In Section 4, some numerical examples are given to verify the numerical properties and parallel performance of the methods with the matrices derived from molecular orbital computation.

2 An eigensolver using a contour integration

In this section, we describe an eigensolver using a contour integration presented in [8]. This method finds eigenpairs inside a given circle. Let $(\lambda_i, \boldsymbol{x}_i)$, $1 \leq i \leq n$ be eigenpairs of the matrix pencil (A, B). Suppose that m eigenvalues $\lambda_1, \ldots, \lambda_m$ are located inside the circle Γ with the center $\gamma \in \mathbb{R}$ and the radius $\rho > 0$ in the complex plane. For a nonzero vector $\boldsymbol{v} \in \mathbb{R}^n$, let

$$\boldsymbol{s}_k = \frac{1}{2\pi \mathrm{i}} \int_\Gamma z^k (zB - A)^{-1} B \boldsymbol{v} \, \mathrm{d}z, \quad k = 0, 1, \ldots, m-1, \tag{1}$$

and let $S = [\boldsymbol{s}_0, \ldots, \boldsymbol{s}_{m-1}] \in \mathbb{R}^{n \times m}$.

Suppose that $\boldsymbol{v}$ is expanded by the eigenvectors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ as

$$\boldsymbol{v} = \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i. \tag{2}$$

Then we have the following theorem([8]).

Theorem 1 *If $\lambda_1, \ldots, \lambda_m$ are distinct and $\alpha_j \neq 0$ for $1 \leq j \leq m$ then*

$$\mathrm{span}\{\boldsymbol{s}_0, \ldots, \boldsymbol{s}_{m-1}\} = \mathrm{span}\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}.$$

Let $Q = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m] \in \mathbb{R}^{n \times m}$ be an orthonormal matrix derived from $S = [\boldsymbol{s}_0, \ldots, \boldsymbol{s}_{m-1}]$. Theorem 1 implies that the eigenpairs $(\lambda_i, \boldsymbol{x}_i)$, $1 \leq i \leq m$ can be extracted by using the Rayleigh-Ritz procedure with the projected matrix pencil $(Q^{\mathrm{T}}AQ, Q^{\mathrm{T}}BQ)$.

We approximate the contour integration (1) via the N-point trapezoidal rule:

$$\hat{\boldsymbol{s}}_k = \frac{1}{N} \sum_{j=0}^{N-1} \left(\frac{\omega_j - \gamma}{\rho} \right)^{k+1} (\omega_j B - A)^{-1} B \boldsymbol{v}, \quad k = 0, 1, \ldots, m-1$$

where $\omega_j = \gamma + \rho \exp(2\pi \mathrm{i}(j + 1/2)/N)$ and N is a positive integer.

In this computation, we solve the systems of linear equations

$$(\omega_j B - A)\boldsymbol{y}_j = B\boldsymbol{v}, \quad j = 0, 1, \ldots, N-1 \tag{3}$$

for $\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{N-1} \in \mathbb{R}^n$. Note that the solutions $\boldsymbol{y}_{N/2}, \ldots, \boldsymbol{y}_{N-1}$ are obtained from the relation $\boldsymbol{y}_j = \bar{\boldsymbol{y}}_{N-j-1}$. Thus we only need to solve $N/2$ systems, and

$$\hat{\boldsymbol{s}}_k = \frac{1}{N} \sum_{j=0}^{N/2-1} 2\,\mathrm{Re}\left(\left(\frac{\omega_j - \gamma}{\rho} \right)^{k+1} \boldsymbol{y}_j \right), \quad k = 0, 1, \ldots. \tag{4}$$

Let $\theta_j = \exp(2\pi \mathrm{i}(j + 1/2)/N)$, then the following relations are hold.

Lemma 2 *Let η be a real number with $|\eta| \neq 1$. For an integer k $(1 \leq k < N)$, the following holds.*

$$\frac{1}{N} \sum_{j=0}^{N-1} \frac{\theta_j^{k+1}}{\theta_j - \eta} = \frac{\eta^k}{1 + \eta^N}. \tag{5}$$

Proof. If $|\eta| < 1$, we have

$$\begin{aligned} \frac{1}{N} \sum_{j=0}^{N-1} \frac{\theta_j^{k+1}}{\theta_j - \eta} &= \frac{1}{N} \sum_{j=0}^{N-1} \frac{\theta_j^k}{1 - \frac{\eta}{\theta_j}} \\ &= \sum_{p=0}^{\infty} \eta^p \frac{1}{N} \sum_{j=0}^{N-1} \theta^{k-p} \\ &= \sum_{q=0}^{\infty} (-1)^q \eta^{Nq+k}. \end{aligned} \tag{6}$$

The last step follows from the fact that

$$\frac{1}{N} \sum_{j=0}^{N-1} \theta_j^p = \begin{cases} (-1)^q & \text{if } p = qN \text{ for } q \in \mathbb{Z} \\ 0 & \text{otherwise} \end{cases}. \tag{7}$$

Similarly, for the case $|\eta| > 1$, we have

$$\begin{aligned}
\frac{1}{N}\sum_{j=0}^{N-1}\frac{\theta_j^{k+1}}{\theta_j-\eta} &= \frac{1}{N}\sum_{j=0}^{N-1}\left(\frac{-1}{\eta}\right)\frac{\theta_j^{k+1}}{1-\frac{\theta_j}{\eta}} \\
&= \sum_{p=0}^{\infty}\left(\frac{-1}{\eta^{p+1}}\right)\left(\frac{1}{N}\sum_{j=0}^{N-1}\theta^{p+k+1}\right) \\
&= \sum_{q=1}^{\infty}(-1)^{q-1}\eta^{-Nq+k}. \qquad (8)
\end{aligned}$$

If follows from (6) and (8), we have the equation (5). □

Theorem 3 *Let $\eta_i = (\lambda_i - \gamma)/\rho$, $1 \le i \le n$, then*

$$\hat{\boldsymbol{s}}_k = \sum_{i=1}^{n}\frac{\alpha_i}{\rho}\cdot\frac{\eta_i^k}{1+\eta_i^N}\,\boldsymbol{x}_i. \qquad (9)$$

Proof. Since

$$(\omega_j B - A)^{-1}B\boldsymbol{x}_i = \frac{1}{\omega_j-\lambda_i}\,\boldsymbol{x}_i,$$

it follows that

$$\begin{aligned}
\hat{\boldsymbol{s}}_k &= \frac{1}{N}\sum_{j=0}^{N-1}\left(\frac{\omega_j-\gamma}{\rho}\right)^{k+1}(\omega_j B - A)^{-1}B\boldsymbol{v} \\
&= \frac{1}{N}\sum_{j=0}^{N-1}\left(\frac{\omega_j-\gamma}{\rho}\right)^{k+1}\sum_{i=1}^{n}\alpha_i(\omega_j B - A)^{-1}B\boldsymbol{x}_i \\
&= \frac{1}{N}\sum_{j=0}^{N-1}\left(\frac{\omega_j-\gamma}{\rho}\right)^{k+1}\sum_{i=1}^{n}\frac{\alpha_i}{\omega_j-\lambda_i}\boldsymbol{x}_i \\
&= \sum_{i=1}^{n}\frac{\alpha_i}{\rho}\left(\frac{1}{N}\sum_{j=0}^{N-1}\frac{\theta_j^{k+1}}{\theta_j-\eta_i}\right)\boldsymbol{x}_i.
\end{aligned}$$

Therefore, from Lemma 2, we have

$$\hat{\boldsymbol{s}}_k = \sum_{i=1}^{n}\frac{\alpha_i}{\rho}\cdot\frac{\eta_i^k}{1+\eta_i^N}\boldsymbol{x}_i.$$

This proves the theorem. □

For vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$, we define the B inner product $\langle\cdot,\cdot\rangle_B$ by

$$\langle \boldsymbol{a}, \boldsymbol{b}\rangle_B = \boldsymbol{a}^{\mathrm{T}}B\boldsymbol{b}.$$

From Theorem 3, we have the following estimation for eigencomponents that are included in $\hat{\boldsymbol{s}}_k$.

`Input` $A, B \in \mathbb{R}^{n\times n}$, $V \in \mathbb{R}^{n\times L}$, N, M, δ

`Output` $(\hat{\lambda}_i, \hat{\boldsymbol{x}}_i)$, $1 \le i \le \hat{m}$

1. Solve $(\omega_j B - A)Y_j = BV$ for Y_j, $j = 0, \ldots, N/2-1$
2. Compute $\hat{S}_k = \sum_{j=0}^{N/2-1} 2\mathrm{Re}\left(\theta_j^{k+1} Y_j\right)$, $k = 0, \ldots, M-1$
3. Perform singular value decomposition $U^{\mathrm{T}}\Sigma W = [\hat{S}_0, \ldots, \hat{S}_{M-1}]$, and find $\hat{m}$ such that $|\sigma_j|/|\sigma_1| \ge \delta$ for $1 \le j \le \hat{m}$
6. Construct an orthonormal basis $\tilde{Q}$ from $\hat{S}(:, 1:\hat{m})$
7. Form $\tilde{A} = \tilde{Q}^{\mathrm{T}} A \tilde{Q}$ and $\tilde{B} = \tilde{Q}^{\mathrm{T}} B \hat{Q}$
8. Compute the eigenpairs $(\hat{\lambda}_i, \hat{\boldsymbol{w}}_l)$, $1 \le l \le \hat{m}$ of the projected pencil $(\tilde{A}, \tilde{B})$
9. Set $\hat{\boldsymbol{x}}_i = \tilde{Q}\hat{\boldsymbol{w}}_i$, $1 \le l \le \hat{m}$

Figure 1: Algorithm of the block method using the contour integration

Theorem 4 *Suppose that the eivenvectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are normalized to be B-orthogonal, i.e., $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle_B = \delta_{ij}$. Then*

$$|\langle \boldsymbol{x}_i, \hat{\boldsymbol{s}}_k \rangle_B| = \begin{cases} \dfrac{\alpha_i}{\rho}\eta_i^k + O(|\eta_i|^{N+k}) & (1 \le i \le m) \\ O(|\eta_i|^{-N+k}) & (m+1 \le i \le n) \end{cases}.$$

From this theorem, we can see that the eigencomponent in $\hat{\boldsymbol{s}}_k$ with respect to λ_i decays exponentially with the ratio $|\eta_i| = |\lambda_i - \gamma|/\rho$, if λ_i is located outside Γ.

A block variant of the method is proposed in [2], which enable us to obtain degenerate eigenvalues. In this method, a matrix $V = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_L] \in \mathbb{R}^{n\times L}$ is used instead of $\boldsymbol{v}$ in (3), where $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_L$ are linearly independent, and positive integer L is a block size. Then the numerical integration (4) is represented as

$$\hat{S}_k = \frac{1}{N}\sum_{j=0}^{N/2-1} 2\,\mathrm{Re}\left(\left(\frac{\omega_j - \gamma}{\rho}\right)^{k+1} Y_j\right) \quad k = 0, 1, \ldots, M-1,$$

with systems of linear equations

$$(\omega_j B - A)Y_j = BV, \quad j = 0, 1, \ldots, N/2-1, \tag{10}$$

where M is a positive integer chosen so that $M \ge m/L$.

For the choice of the size of the subspace, we use the singular value of $\hat{S} = [\hat{S}_0, \ldots, \hat{S}_{M-1}]$. Let $U\Sigma W = \hat{S}$ be a singular value decomposition of $\hat{S}$, and let $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_M)$. Let $\hat{m}$ be an integer such that $\sigma_j/\sigma_1 \ge \delta$ for $1 \le j \le \hat{m}$ and $\sigma_j/\sigma_1 < \delta$ for $j > \hat{m}$. Then the orthonormal matrix $\tilde{Q} \in \mathbb{R}^{n\times\hat{m}}$ is constructed from $\hat{S}(:, 1:\hat{m})$. The algorithm is shown in Figure 1. Note that we can use $U(:, 1:\hat{m})$ instead of $\tilde{Q}$ to avoid the orthogonalize process of $\hat{S}(:, 1:\hat{m})$.

3 An implementation

When matrices A and B are large, the computational costs for solving systems of linear equations (10) are dominant in our method. In our application, matrices have relatively large number of nonzero elements, which could make sparse direct methods impractical due to large number of fill-ins. Instead, we apply a Krylov subspace iterative method with a complete factorization preconditioner presented in [5].

In our preconditioning method, a complete factorization of the approximate matrix $\tilde{C}_j = (\tilde{c}_{ij})$ of the coefficient matrix $C_j := \omega_j B - A$ is performed. The approximate matrix $\tilde{C}_j$ is obtained from drop-thresholding of the original coefficient matrix C_j. Due to less nonzero entries in $\tilde{C}_j$ than C_j, we expect fewer nonzeroes in the preconditioner than the matrix factor obtained from complete factorization of C_j. The drop-thresholding is defined as follows:

$$\tilde{c}_{ij} = \begin{cases} c_{ij} & (|c_{ij}| > \varepsilon) \\ 0 & (|c_{ij}| \leq \varepsilon) \end{cases},$$

where ε is a small positive number.

Since systems (10) can be solved independently for j, we solve $N/2$ systems

$$C_j Y_j = BV, \quad j = 0, 1, \ldots, N/2 - 1 \tag{11}$$

on each computing node of clusters. Before starting the iterative process of the Krylov subspace method, the approximate matrix $\tilde{C}_j$ is factorized, and then forward/backward substitutions and matrix-vector multiplies are performed in the iterative process.

We can easily extend the method for the case that several number of circular regions are given. Suppose that N_c circles $\Gamma^{(0)}, \ldots, \Gamma^{(N_C-1)}$ with center and radius $(\gamma^{(l)}, \rho^{(l)})$, $0 \leq l \leq N_C - 1$ are given. Then we solve $N_C \times (N/2)$ linear systems

$$(\omega_j^{(l)} B - A) Y_j^{(l)} = BV, \quad j = 0, \ldots, N/2 - 1, \quad l = 0, \ldots, N_C - 1,$$

where $\omega_j^{(l)} = \gamma^{(l)} + \rho^{(l)} \exp(2\pi \mathrm{i}(j + 1/2)/N)$, $j = 0, \ldots, N/2 - 1$ are equidistributed points on the l-th circle $\Gamma^{(l)}$. We solve these linear systems simultaneously on distributed computing nodes.

In the beginning, we broadcast the sparse matrix data of A and B to all computing nodes. Then we start to solve systems of linear equations on the computing nodes simultaneously. After that, we construct a subspace for each circle using the results of the linear solver, and then perform the Rayleigh-Ritz procedure to extract eigenpairs in each circle. The computational costs to solve systems of linear equations are dominant in whole computations, thus we can expect high performance in parallel computations. The algorithm is shown in Figure 2.

1. `Broadcast` sparse matrix data of A and B
2. `do` $l = 0, 1, \ldots, N_C - 1$
3. Solve $(\omega_j^{(l)} B - A) Y_j^{(l)} = BV$ for $Y_j^{(l)}$, $j = 0, \ldots, N/2 - 1$
4. `AllReduce` $2\mathrm{Re}(\theta_j^{(l)})^{k+1} Y_j^{(l)}$, $j = 0, \ldots, N/2 - 1$, and compute $\hat{S}_k^{(l)} = \sum_{j=0}^{N/2-1} 2\mathrm{Re}\left((\theta_j^{(l)})^{k+1} Y_j^{(l)}\right)$, $k = 0, \ldots, M - 1$
5. Compute singular values $\sigma_1^{(l)}, \ldots, \sigma_M^{(l)}$ of $\hat{S}^{(l)} = [\hat{S}_0^{(l)}, \ldots, \hat{S}_{M-1}^{(l)}]$, and set $\hat{m}^{(l)}$ such that $|\sigma_j^{(l)}|/|\sigma_1^{(l)}| \geq \delta$ for $1 \leq j \leq \hat{m}^{(l)}$
6. Construct the orthnormal basis $\tilde{Q}^{(l)}$ from $\hat{S}^{(l)}(:, 1 : \hat{m}^{(l)})$
7. Form $\tilde{A}^{(l)} = (\tilde{Q}^{(l)})^{\mathrm{T}} A \tilde{Q}^{(l)}$ and $\tilde{B}^{(l)} = (\tilde{Q}^{(l)})^{\mathrm{T}} B Q^{(l)}$
8. Compute the eigenpairs $(\hat{\lambda}_i^{(l)}, \hat{\boldsymbol{w}}_i^{(l)})$, $1 \leq i \leq \hat{m}^{(l)}$ of the projected pencil $(\tilde{A}^{(l)}, \tilde{B}^{(l)})$
9. Set $\hat{\boldsymbol{x}}_i^{(l)} = \tilde{Q}^{(l)} \hat{\boldsymbol{w}}_i^{(l)}$, $1 \leq i \leq \hat{m}^{(l)}$
10. `end do`

Figure 2: A parallel implementation with circles $\Gamma_0, \ldots, \Gamma_{N_C-1}$

4 Numerical examples

In this section, we show some numerical examples with the matrices that are derived from the FMO-MO method[10].

Example 1 *The test matrices A and B are derived from computation of the molecular orbitals of an eight-base-pair model DNA. The size of matrices is $n = 1,980$, and the number of nonzero elements is $728,080$.*

The CPU was Core2Duo(2.2GHz) with 2GB of memory. Computation was performed in MATLAB 7.5 with double-precision arithmetic. The systems of linear equations were solved by the sparse direct solver UMFPACK in MATLAB. The elements of V were distributed randomly on the interval $[0, 1]$ by a random number generator. The exact eigenpairs $(\lambda_i, \boldsymbol{x}_i)$ $1 \leq i \leq n$ were evaluated by the MATLAB command `eig`.

In Figure 3, we show $|\langle \boldsymbol{x}_i, \hat{\boldsymbol{s}}_0 \rangle_B|$ with respect to eigenvalues $\lambda_i \in [-0.25, -0.16]$. The parameters were $\gamma = -0.2$, $\rho = 0.01$, $N = 32$, $L = 12$, $M = 12$, and $\delta = 10^{-12}$. In this case, the interval $[-0.21, -0.19]$ is inside and on Γ. We can see that the eigencomponents associated with eigenvalues that are located outside Γ decay exponentially to zero in $\hat{\boldsymbol{s}}_0$. For vectors $\hat{\boldsymbol{s}}_k$, $k \geq 1$, we can obtain similar results. In Figure 4, we show $|\langle \hat{\boldsymbol{x}}_i, \tilde{Q} \rangle_B|$. We can see that $\tilde{Q}$ includes only the eigencomponents around the interval $[-0.21, -0.19]$.

In Table 1, the error and residual of approximate eigenvalues in Γ are shown. 16 eigenvalues are located in the interval $[-0.21, -0.19]$, and all the eigenvalues in this interval were obtained with sufficient accuracy.

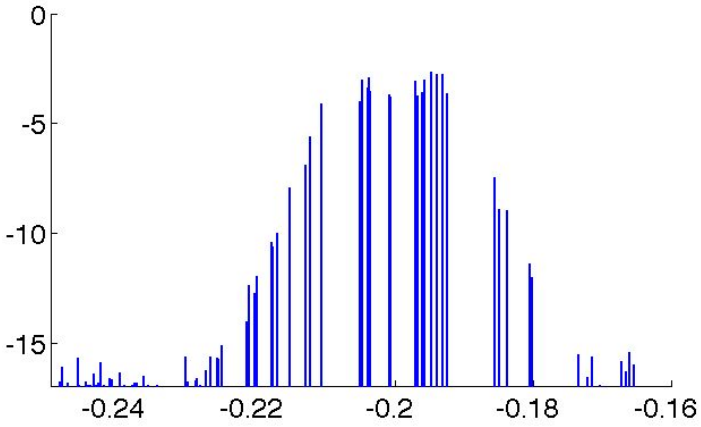

Figure 3: $|\langle \boldsymbol{x}_i, \hat{\boldsymbol{s}}_0 \rangle_B|$ ($\gamma = -0.2$, $\rho = 0.01$, $N = 32$)

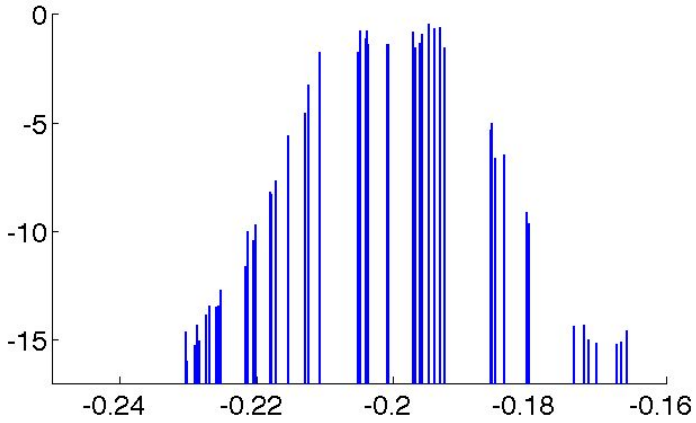

Figure 4: $|\langle \tilde{\boldsymbol{x}}_i, \tilde{Q} \rangle_B|$ ($\gamma = -0.2$, $\rho = 0.01$, $N = 32$)

Example 2 *The test matrices were obtained from an epidermal growth factor receptor (EGFR) protein which is known as a target molecule for anticancer agent. The Fock matrix was constructed by the method described in [10]. The dimension of the Fock matrix was* $96,234$*, and the number of non-zero elements was* 457 *million.*

The calculations were performed on P32 subsystem of the AIST Super Cluster (ASC) in National Institute of Advanced Industrial Science and Technology Agency (AIST). Each computing node is Opteron Dual processor (2.0GHz) with 6 GB of memory, and is interconnected by both Myrinet and gigabit ethernet. Intel C and Fortran compiler 9.1 were used to compile the codes with Intel Math Kernel Library 10.0.

The number of integration points was $N = 32$, and the block size was $L = 16$. We put 8 circles around the highest occupied molecular orbital (HOMO) and the lowest unoccupied molecular orbital (LUMO) using the approximate eigenvalues obtained from the results of the FMO method. Then the total number of systems to solve was $8 \times (32/2) = 128$ with 16 right-hand side vectors. Each system of linear equations was solved on single node of dual processor, thus 16 nodes (32 PUs) were used for one circle. Therefore total 256 PUs were employed.

Table 1: Errors and residuals in Example 1.

i	$\hat{\lambda}_i$	$\lvert\hat{\lambda}_i - \lambda_i\rvert$	$\lVert A\hat{\boldsymbol{x}}_i - \hat{\lambda}_i B\hat{\boldsymbol{x}}_i\rVert_2$
1	-0.192423480937650	2.9×10^{-15}	9.4×10^{-15}
2	-0.193053272393176	3.7×10^{-15}	1.6×10^{-14}
3	-0.193867518039814	1.1×10^{-15}	1.0×10^{-14}
4	-0.194749418147652	2.0×10^{-14}	2.0×10^{-14}
5	-0.195768218806505	2.5×10^{-15}	2.5×10^{-14}
6	-0.196048479728466	7.2×10^{-16}	2.8×10^{-14}
7	-0.196721896535061	1.2×10^{-15}	4.0×10^{-14}
8	-0.197042777317239	4.7×10^{-16}	5.5×10^{-14}
9	-0.200591957877910	1.2×10^{-15}	5.5×10^{-13}
10	-0.200835717839500	5.8×10^{-16}	1.9×10^{-13}
11	-0.203655629362533	2.5×10^{-16}	5.6×10^{-14}
12	-0.203800310807316	1.4×10^{-16}	4.4×10^{-14}
13	-0.203932047603002	6.1×10^{-16}	4.4×10^{-14}
14	-0.204805988179016	3.4×10^{-15}	1.6×10^{-14}
15	-0.204824251443765	2.6×10^{-15}	3.4×10^{-14}
16	-0.205054724756940	1.3×10^{-14}	2.8×10^{-14}

The preconditioned COCG method[11] was used for iterative linear solver. The stopping criterion for the relative residual norm was 10^{-10}. The preconditioner was constructed by applying a complete factorization for an approximate coefficient matrix which was obtained from drop-thresholding of the original coefficient matrix. The drop-thresholding parameter was 2×10^{-4}. The complete factorization is performed by a sparse direct solver in the PARDISO library[9]. The elements of V were distributed randomly on the interval $[0, 1]$ by a random number generator.

The timing results were shown in Figure 5. The wall-clock time was 602 seconds, and 94 eigenpairs were obtained. The time to broadcast the matrix data to all computing nodes was 36.4 seconds, and the time to solve linear systems was 563.8 seconds. The maximum residual of these eigenpairs was 3.4×10^{-10}.

Figure 6 shows the amount time of each step in the algorithm for the circle $\Gamma^{(7)}$, where gray, light gray, white, and black parts represent the time for factorization of approximate matrix, forward and backward substitution in iteration, matrix-matrix multiply, and others, respectively.

5 Conclusions

A parallel method for computing interior eigenvalues and corresponding eigenvectors of generalized eigenvalue problems which arise from molecular orbital computation was considered. In our application, matrices often have relatively large number of

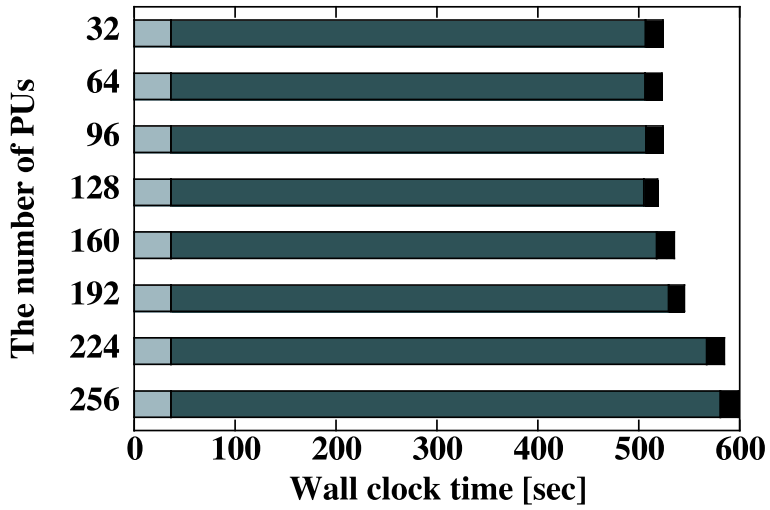

Figure 5: Timing results of the method with 256 PUs in seconds. Each bar represents the wall-clock time for one circle. 32 PUs were used for each circle, and total 256 PUs were used. (■: Broadcast, ■: Linear system solver ■: Rayleigh-Ritz procedure)

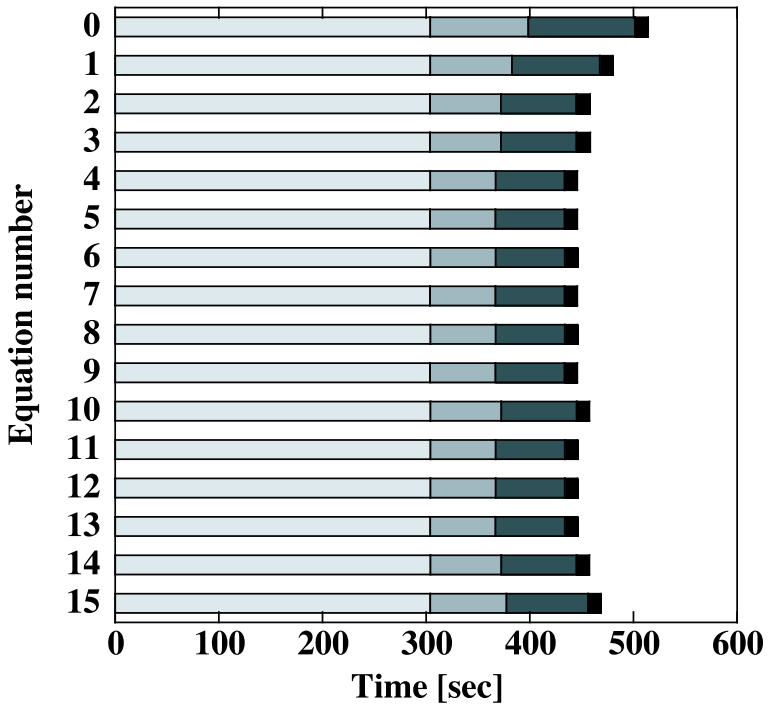

Figure 6: Amount time of each process for one circle. Each bar represents the time on one integration node. 16 systems of linear equations were solved. (■: Factorization, ■: Forward/backward Substitution, ■: Matrix-vector multiply, ■: Rayleigh-Ritz procedure)

nonzero elements, and we require some eigenpairs in a specific part of the spectrum.

We applied a Rayleigh-Ritz type method with numerical integration. In this method, a contour integration is used to construct a desired subspace. Our eigensolver does not require inner/outer loops to construct an approximate subspace and update approximate eigenvectors. For the computation of the contour integration, we solve a certain number of systems of linear equations. When A and B are large, the computational costs for solving these systems of linear equations are dominant. Since these linear systems can be solved independently for each integration node, the process to derive the subspace is performed in parallel.

We showed that the subspace obtained by a numerical integration includes eigenvectors corresponding to the eigenvalues around the interested interval. The i-th eigencomponent in the subspace decays exponentially to zero with a rate $|\lambda_i - \gamma|/\rho$, if the corresponding eigenvalue λ_i is located outside the interval.

Through the numerical experiments, we can see that the presented method gives good parallel performance, and we can obtain desired eigenpairs with sufficient accuracy. Estimation of appropriate parameters with theoretical analysis and comparison with other methods are part of our future work.

Acknowledgements

This work was supported in part by CREST of the Japan Science and Technology Agency (JST).

References

[1] T. Ikegami, T. Ishida, D. G. Fedorov, K. Kitaura, Y. Inadomi, H. Umeda, M. Yokokawa and S. Sekiguchi, Full electron calculation beyond 20,000 atoms: ground electronic state of photosynthetic proteins, Proceedings of Supercomputing 2005, Seattle, Washington, USA. 2005.

[2] T. Ikegami, T. Sakurai, U. Nagashima, A filter diagonalization for generalized eigenvalue problems based on the Sakurai-Sugiura projection method, Technical Report CS-TR-08-13, Tsukuba, 2008.

[3] Y. Inadomi, T. Nakano, K. Kitaura, and U. Nagashima, Definition of molecular orbitals in fragment molecular orbital method, *Chem. Phys. Lett.* 364 (2002) 139–143.

[4] K. Kitaura, T. Sawai, T. Asada, T. Nakano and M. Uebayasi, Pair interaction molecular orbital method: an approximate computational method for molecular interactions. *Chem. Phys. Lett.* **312** (1999) 319–324.

[5] M. Okada, T. Sakurai, K. Teranishi, A preconditioner for krylov subspace method using a sparse direct solver. Abstract of Precond'07, Toulouse, 2007.

[6] T. Sakurai and H. Sugiura, A projection method for generalized eigenvalue problems, *J. Comput. Appl. Math.* **159** (2003) 119–128.

[7] T. Sakurai, Y. Kodaki, H. Tadano, H. Umeda, Y. Inadomi, T. Watanabe, U. Nagashima, A master-worker type eigensolver for molecular orbital computations. Kagstrom, B., Elmroth, E., Dongarra, Wasniewski, J. (eds.) PARA'06. LNCS No. 4699, 617–625, Springer, Heidelberg (2007).

[8] T. Sakurai, T. Tadano, CIRR: a Rayleigh-Ritz type method with contour integral for generalized eigenvalue problems, In: The First China-Japan-Korea Joint Conference on Numerical Mathematics, *Special Issue of Hokkaido Math. J.* **36** (2007) 745–757.

[9] O. Schenk and K. Gärtner, Solving unsymmetric sparse systems of linear equations with PARDISO, *Future Generation Comput. Sys.* **20** (2004) 475–487.

[10] H. Umeda, Y. Inadomi, T. Watanabe, T. Ishimoto, U. Nagashima, Grid-enabled parallel Fock matrix construction, Abstract of MATH/CHEM/COMP 2007 (MCC'07), Dubrovnik, Croatia, 2007.

[11] H. A. van der Vorst and J. B. M. Melissen, A Petrov-Galerkin type method for solving $Ax = b$, where A is a symmetric complex matrix, *IEEE Trans. on Magnetics* **26** (1990) 706–708.

[12] T. Watanabe, Y. Inadomi, T. Ishimoto, H. Umeda, T. Sakurai, and U. Nagashima, Molecular orbital calculation for large molecule, *J. Comput. Chem. Japan*, **6** (2007) 217–226.