Journal of Computational Mathematics, Vol.18, No.4, 2000, 387–402.

A PROCESS FOR SOLVING A FEW EXTREME EIGENPAIRS OF LARGE SPARSE POSITIVE DEFINITE GENERALIZED EIGENVALUE PROBLEM *

Chong-hua Yu
(*Department of Mathematics, Fudan University, Shanghai 200433, China*)
O. Axelsson
(*Faculty of Mathematics and Informatics, University of Nijmegen, 6525 ED Nijmegen, The Netherlands*)

Abstract

In this paper, an algorithm for computing some of the largest (smallest) generalized eigenvalues with corresponding eigenvectors of a sparse symmetric positive definite matrix pencil is presented. The algorithm uses an iteration function and inverse power iteration process to get the largest one first, then executes $m-1$ Lanczos-like steps to get initial approximations of the next $m-1$ ones, without computing any Ritz pair, for which a procedure combining Rayleigh quotient iteration with shifted inverse power iteration is used to obtain more accurate eigenvalues and eigenvectors. This algorithm keeps the advantages of preserving sparsity of the original matrices as in Lanczos method and RQI and converges with a higher rate than the method described in [12] and provides a simple technique to compute initial approximate pairs which are guaranteed to converge to the wanted m largest eigenpairs using RQI. In addition, it avoids some of the disadvantages of Lanczos and RQI, for solving extreme eigenproblems. When symmetric positive definite linear systems must be solved in the process, an algebraic multilevel iteration method (AMLI) is applied. The algorithm is fully parallelizable.

Key words: Eigenvalue, sparse problem

1. Introduction

We are concerned in this work with finding a few extreme eigenvalues and their corresponding eigenvectors of a generalized large scale eigenvalue problem in which the matrices are sparse and symmetric positive definite.

Although finding a few extreme eigenpairs is of interest both in theory and practice, there are only few usable and efficient methods up to now. Reinsch and Bauer ([12]), suggested a QR algorithm with Newton shift for the standard eigenproblem which included an ingenious method to evaluate the shift. Their algorithm has a lower asymptotic convergence rate than a normal QL process with Wilkinson's shift or Rayleigh quotient iteration with inverse power iteration (RQI) and their strategy to calculate the shift can not be extended to the situation where any other shift which is different from the Newton shift and with a higher convergence rate is used. There are some works for selecting different shifts and acceleration techniques, see [11], [16], [17], [18].

* Received April 28, 1997.

There is lack of efficient methods of computing extreme eigenvalues, in particular for the generalized problem. Although in theory it can be transformed into a standard problem as long as one of the two matrices A and B is positive definite, such a transformation will destroy the sparse structure of the initial matrices and is therefore inapplicable in practice.

Lanczos-like method and shifted inverse power iteration combined with Rayleigh quotient shift (or other evaluable shift) are expected to have a good efficiency for sparse eigenvalue problem either a standard one or a generalized one because it will not destroy the sparsity of matrices and the RQI method has a higher convergence rate, see [7], [3], [9]. But there exist several difficulties when using such a method to an extreme eigenvalue problem. First, it seems to be impossible to decide how many steps must be executed in the Lanczos process for getting the m largest eigenvalues. This means one must perform many more Lanczos steps than m and it still does not guarantee obtaining enough extreme eigenpairs as required. Another problem is that it is not certain whether it will converge to those expected m largest eigenpairs during the RQI process in which such Ritz pairs are taken as initial values.

We suggest an algorithm in Section 2 which avoids the above two weak points, i.e., it produces just m original pairs, without solving the Ritz pairs of the matrix pencil, and which will be guaranteed to tend to the m largest needed eigenpairs and even with a higher rate. Furthermore it also preserves the sparse structure of the original matrices and is fully parallelizable.

A sequence of shifts $\{\sigma_k\}$, which converges to the largest generalized eigenvalue of matrix pencil is computed first, by using an iteration function and then the largest eigenpair is computed by an inverse power iteration process. Subsequently, $m-1$ steps of a Lanczos-like procedure from this pair are performed to compute $m-1$ values and vectors which are good approximate extreme generalized eigenpairs, as will be proved in Section 3. More precisely, instead of solving the Ritz pairs of the original matrices, we could take them as initial pairs immediately and get the required m largest eigenpairs by Rayleigh quotient iteration and inverse power iteration which, furthermore, will converge with a cubic rate asymptotically since each original pair is very close to the corresponding final pair.

We present some further theoretical analysis in Section 3. In Section 4, we discuss some practical computational aspects, such as how to construct the iteration function, how to calculate function values and derivative values of the eigenpolynomial, how to solve the corresponding linear system and parallelization aspects, etc. Some numerical examples and concluding remarks are found in the two final sections.

2. Algorithm

Consider the generalized eigenvalue problem

$$A\mathbf{z} = \lambda B\mathbf{z} \tag{1}$$

where both A and B are $n \times n$ sparse real symmetric matrices and in addition B is positive definite. Let the n eigenpairs be $(\lambda_1, \mathbf{z}_1)$, $(\lambda_2, \mathbf{z}_2)$,...,$(\lambda_n, \mathbf{z}_n)$. We assume

$$(B\mathbf{z}_i, \mathbf{z}_j) = \delta_{ij}$$

i.e., $\{\mathbf{z}_i\}$ is a B-orthogonal basis and $\|\mathbf{z}_i\|_B = 1$, where the norm $\|\cdot\|_B = \sqrt{(B\cdot, \cdot)}$ and

$$\lambda_1 > \lambda_2 > \ldots > \lambda_n \tag{2}$$

(We will discuss the situation, where some equalities are attainable, in another paper. We point out here that it is not important in a practical computational procedure although this assumption seems to put a severe restriction in theory. See Example 3 in Section 5.)

Our task is finding the m largest eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$ and corresponding eigenvectors $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m$, where $m \ll n$. (Computing the m smallest ones is similar.)

The proposed algorithm consists of the following steps:

Algorithm 1. First, we select an iterative function $f(x)$ and an initial value $\sigma_1 > \lambda_1$, such that the iteration

$$\sigma_{k+1} = f(\sigma_k), \quad k = 1, 2, \ldots \tag{3}$$

can produce a sequence $\{\sigma_k\}$ which satisfies $\sigma_k > \lambda_1$ for all k and

$$\lim_{k\to\infty} \sigma_k = \lambda_1 \tag{4}$$

with rate of convergence p. (How to construct $f(x)$ will be discussed in Section 4.)

Step 1: Choose σ_1 and $\mathbf{u}_1$, $\|\mathbf{u}_1\|_B = 1$. Let $k = 1$,

Step 2: Inverse power iterations with shift $\{\sigma_k\}$

(a) Solve $(\sigma_k B - A)\tilde{\mathbf{u}}_{k+1} = -B\mathbf{u}_k$

(b) $\mathbf{u}_{k+1} = \tilde{\mathbf{u}}_{k+1}/\|\tilde{\mathbf{u}}_{k+1}\|_B$, $\sigma_{k+1} = f(\sigma_k)$

(c) IF $\|\tilde{\mathbf{u}}_{k+1}\|_B^{-1} > \varepsilon_1$ AND $|\sigma_{k+1} - \sigma_k| > \varepsilon_2|\sigma_{k+1}|$ THEN

$k = k + 1$, go to Step 2 (a)

ELSE

$\mathbf{v}_1 = \mathbf{u}_{k+1}$, $\mu_1 = (A\mathbf{v}_1, \mathbf{v}_1)$

END

Step 3: $a_1 = \mu_1$, $b_0 = 0$, $\mathbf{q}_1 = \mathbf{v}_1$

FOR $i = 1, 2, \ldots, m-1$

Solve $B\mathbf{r} = A\mathbf{q}_i$

$\mathbf{r} = \mathbf{r} - a_i\mathbf{q}_i - b_{i-1}\mathbf{q}_{i-1}$

$b_i = \sqrt{(B\mathbf{r}, \mathbf{r})}$, $\mathbf{q}_{i+1} = \mathbf{r}/b_i$, $a_{i+1} = (A\mathbf{q}_{i+1}, \mathbf{q}_{i+1})$

END

Step 4: Execute iteration combined Rayleigh quotient and inverse power iteration with $(a_i, \mathbf{q}_i)$ as initial approximation to get a more accurate approximation of $(\lambda_i, \mathbf{z}_i)$:

FOR $i = 2, 3, \ldots, m$

$\sigma_1 = a_i$ and $\mathbf{u}_1 = \mathbf{q}_i$. $k = 1$

(a) Solve $(A - \sigma_k B)\tilde{\mathbf{u}}_{k+1} = B\mathbf{u}_k$

(b) $\mathbf{u}_{k+1} = \tilde{\mathbf{u}}_{k+1}/\|\tilde{\mathbf{u}}_{k+1}\|_B$, $\sigma_{k+1} = (A\mathbf{u}_{k+1}, \mathbf{u}_{k+1})$

(c) IF $\|\tilde{\mathbf{u}}_{k+1}\|_B^{-1} > \varepsilon_1$ AND $|\sigma_{k+1} - \sigma_k| > \varepsilon_2|\sigma_{k+1}|$ THEN

$k = k + 1$, go to Step 4 (a)

ELSE

$\mathbf{v}_i = \mathbf{u}_{k+1}$,$\mu_i = \sigma_{k+1}$

END

END

Thus we get $(\mu_1, \mathbf{v}_1)$, $(\mu_2, \mathbf{v}_2)$, ..., $(\mu_m, \mathbf{v}_m)$ after Step 4, which are taken as $(\lambda_1, \mathbf{z}_1)$, $(\lambda_2, \mathbf{z}_2)$, ..., $(\lambda_m, \mathbf{z}_m)$.

Remark 2.1. In the next section, we will prove that after inverse power iteration with shift $\{\sigma_k\}$ as in Step 2 the B-orthogonal vector $\mathbf{q}_i$ and diagonal coefficient a_i are

good approximates of $(\lambda_i, \mathbf{z}_i)$, the i-th largest eigenpair of (1). So there is no need to compute its Ritz pairs, i.e., the tridiagonal matrix eigenpairs although Step 3 is a tridiagonalization like Lanczos method (some further statements could be found in Section 6).

Remark 2.2. Step 4 is a normal Rayleigh quotient and inverse iteration from $\{(a_i, \mathbf{q}_i)\}$. It is well known (see [11], for instance) that if $\{(a_i, \mathbf{q}_i)\}$ are sufficiently accurate approximates of $\{(\lambda_i, \mathbf{z}_i)\}$, then the RQI will converge with cubic speed.

Remark 2.3. It's clear that this algorithm keeps sparse structure of the original matrices.

Remark 2.4. ε_1 in (c) of step 2 and 4 is the prescribed precision for the eigenvector. The eigenvalue is typically "square" as accurate as the eigenvector (see [11], for example) so we always use $\varepsilon_2 = \varepsilon_1^2$ as the stopping criterion for the eigenvalue.

3. Theoretical Analysis

To simplify the discussion, we change first (1) into an equivalent standard eigenvalue problem.

Let the Cholesky factorization of the symmetric positive definite matrix B be

$$B = LL^T$$

then

$$(L^{-1}AL^{-T})(L^T\mathbf{z}) = \lambda(L^T\mathbf{z}).$$

Writing $M = L^{-1}AL^{-T}$ and $\mathbf{x} = L^T\mathbf{z}$, we get

$$M\mathbf{x} = \lambda\mathbf{x} \tag{5}$$

whose n eigenpairs are $(\lambda_1, \mathbf{x}_1)$, $(\lambda_2, \mathbf{x}_2)$,...,$(\lambda_n, \mathbf{x}_n)$, where $\mathbf{x}_i = L^T\mathbf{z}_i$ for all i.

Now we consider the following algorithm which corresponds to Algorithm 1.

Algorithm 2. Step 1′: Choose σ_1 and $\mathbf{w}_1 = L^T\mathbf{u}_1$, (so $\|\mathbf{w}_1\| = 1$). $k = 1$

Step 2′: (a) Solve $(M - \sigma_k I)\tilde{\mathbf{w}}_{k+1} = \mathbf{w}_k$

(b) $\mathbf{w}_{k+1} = \tilde{\mathbf{w}}_{k+1}/\|\tilde{\mathbf{w}}_{k+1}\|$, $\sigma_{k+1} = f(\sigma_k)$

(c) IF $\|\tilde{\mathbf{w}}_{k+1}\|^{-1} > \varepsilon_1$ AND $|\sigma_{k+1} - \sigma_k| > \varepsilon_2|\sigma_{k+1}|$ THEN

$k = k + 1$, go to Step 2′ (a)

ELSE

$\mathbf{y}_1 = \mathbf{w}_{k+1}$, $\nu_1 = (M\mathbf{y}_1, \mathbf{y}_1)$,

END

Step 3′: $\rho_1 = \nu_1$, $\pi_0 = 0$, $\mathbf{p}_1 = \mathbf{y}_1$

FOR $i = 1, 2, \ldots, m-1$

$\mathbf{r} = M\mathbf{p}_i - \rho_i\mathbf{p}_i - \pi_{i-1}\mathbf{p}_{i-1}$

$\pi_i = \sqrt{(\mathbf{r}, \mathbf{r})}$, $\mathbf{p}_{i+1} = \mathbf{r}/\pi_i$, $\rho_{i+1} = (M\mathbf{p}_{i+1}, \mathbf{p}_{i+1})$

END

We omit step 4 for the reason stated in Remark 2.2.

It follows readily by induction that

In the two algorithms in Section 2 and 3, there exist the relation

$$\mathbf{w}_k = L^T\mathbf{u}_k, \quad \mathbf{p}_i = L^T\mathbf{q}_i, \quad a_i = \rho_i, \quad b_i = \pi_i$$

so we need only prove the following:

(i) $\mathbf{w}_k \to \mathbf{x}_1$, with some rate p as $k \to \infty$,

(ii) $(\rho_i, \mathbf{p}_i)$ is a better approximate of $(\lambda_i, \mathbf{x}_i)$, for $i = 2, 3, \ldots, m$.

Using The Uniqueness of Reduction Theorem (see [11]), we can find a symmetric tridiagonal matrix T and an orthogonal matrix W, whose first column is $\mathbf{w}_1$, such that

$$MW = WT \tag{6}$$

Without loss of generality, we assume that T is irreducible, i.e. T's subdiagonal elements are non-zero due to (2); otherwise we can perturbe them by a small enough positive number instead of zero subdiagonal elements.

Let $T^{(1)} \equiv T$ and execute the QL process with shift $\{\sigma_k\}$ which is the same as in step $2'$

$$\begin{cases} T^{(k)} - \sigma_k I = Q_k L_k, \\ T^{(k+1)} = L_k Q_k + \sigma_k I, \quad k = 1, 2, \ldots \end{cases} \tag{7}$$

where

$$T^{(k)} \equiv \mathrm{tridiag}[\beta_{\mathrm{i-1}}^{(\mathrm{k})}, \alpha_{\mathrm{i}}^{(\mathrm{k})}, \beta_{\mathrm{i}}^{(\mathrm{k})}] = \begin{bmatrix} \alpha_1^{(k)} & \beta_1^{(k)} & & \\ \beta_1^{(k)} & \alpha_2^{(k)} & \beta_2^{(k)} & \\ & \ddots & \ddots & \ddots \\ & & \beta_{n-1}^{(k)} & \alpha_n^{(k)} \end{bmatrix}, (\beta_i^{(k)} > 0)$$

Q_k is an orthogonal and L_k is a lower triangular matrix with positive diagonal elements. Such an orthogonal-triangular factorization is unique.

Now we build a relationship between Algorithm 2 and the QL process.

Lemma 1. *There exist an orthogonal matrix P_k such that*

$$MP_k = P_k T^{(k+1)} \tag{8}$$

and

$$P_k \mathbf{e}_i = \mathbf{p}_i, \quad \rho_i = \alpha_i^{(k+1)}, \quad (i = 1, 2, \ldots, m), \quad \pi_i = \beta_i^{(k+1)}, \quad (i = 1, 2, \ldots, m-1)$$

here $\mathbf{p}_i$, ρ_i, π_i are denoted in Algorithm 2 and $\mathbf{e}_i$ is the i-th column of an identity matrix.

Proof. Let the matrix sequence $\{P_k\}$ be defined by

$$\begin{cases} P_0 = W \\ P_k = P_{k-1} Q_k = W Q_1 Q_2 \cdots Q_k, \quad k = 1, 2, \ldots \end{cases}$$

from which it follows that

$$MP_k = P_k T^{(k+1)}$$

For $k = 1$, by (7) it follows

$$(T^{(1)} - \sigma_1 I) Q_1 \mathbf{e}_1 = L_1^T \mathbf{e}_1.$$

i.e.

$$(M - \sigma_1 I) W Q_1 \mathbf{e}_1 = W L_1^T \mathbf{e}_1$$

and by (8),

$$(M - \sigma_1 I)(P_1 \mathbf{e}_1) = l_{11}^{(1)} (W \mathbf{e}_1), \quad (\|P_1 \mathbf{e}_1\| = 1, l_{11} > 0)$$

which we compare with step $2'$,

$$(M - \sigma_1 I)\mathbf{w}_2 = \|\tilde{\mathbf{w}}_2\|^{-1}\mathbf{w}_1, \quad (\|\mathbf{w}_2\| = 1).$$

Since σ_1 is not an eigenvalue of M, we have

$$P_1\mathbf{e}_1 = \mathbf{w}_2.$$

By induction, we have

$$P_k\mathbf{e}_1 = \mathbf{w}_{k+1} = \mathbf{p}_1.$$

Executing the Lanczos process to transform M by orthogonal mappings into a symmetric tridiagonal form from the original vector $\mathbf{p}_1$ and using (8), it is readily seen that

$$P_k\mathbf{e}_i = \mathbf{p}_i, \quad \rho_i = \alpha_i^{(k+1)}, \quad (i = 1, 2, \ldots, m)$$

and

$$\pi_i = \beta_i^{(k+1)}, \quad (i = 1, 2, \ldots, m-1) \qquad \square$$

Now, we give the main theorem on our algorithm as follows:

Theorem 1. *If $\sigma_k \to \lambda_1$, the largest eigenvalue of M, with rate p ($k \to \infty$) in Algorithm 2, then*

(i) $\mathbf{p}_1 \to \mathbf{x}_1$, $\nu_1 \to \lambda_1$, $\pi_1 \to 0$, *(with rate p).*
(ii) $(\rho_i, \mathbf{p}_i) \to (\lambda_i, \mathbf{x}_i)$, *for $i = 2, 3, \ldots, m$,*

By this theorem, we know that $(\rho_i, \mathbf{p}_i)$ is a better approximate of $(\lambda_i, \mathbf{x}_i)$ for $i = 2, 3, \ldots, m$, as $(\nu_1, \mathbf{y}_1)$ converges to $(\lambda_1, \mathbf{x}_1)$, the largest eigenpair, in step $2'$.

To prove this theorem, we need the following two lemmas which are provided without proofs. They can be found in [16] and proofs could be found in [17], [19] and [9].

Lemma 2. *Let $\tilde{X}$ be an $n \times n$ matrix whose n columns are T's n different eigenvectors arranged by arbitrary order, then the first $k \times k$ principal submatrices $\tilde{X}_k$, $k = 1, 2, \ldots, n$, have nonzero determinants.*

Lemma 3. *If $\sigma_k \to \lambda_i$ with rate p and QL process with shift $\{\sigma_k\}$ converges by speed q as well as $\beta_2^{(k)} \to 0$, then $p = q$.*

Now we give a proof of Theorem 1.

Proof. At first, we prove that

$$T^{(k+1)} \to \Lambda = \mathrm{diag}[\lambda_1, \lambda_2, \ldots, \lambda_n], \quad (k \to \infty) \tag{9}$$

Let the Jordan decomposition of matrix $T \equiv T^{(1)}$ be

$$T = X\Lambda X^{-1} \tag{10}$$

By Lemma 2, every principal minor determinant of X is nonzero, so X can be decomposed in a Crout form as

$$X = LR,$$

where L is a lower triangular matrix and R is an upper triangular matrix with diagonal elements 1. Denote

$$\Lambda_k = \mathrm{diag}[\prod_{l=1}^{k}(\lambda_1 - \sigma_l), \prod_{l=1}^{k}(\lambda_2 - \sigma_l), \ldots, \prod_{l=1}^{k}(\lambda_n - \sigma_l)]$$

then

$$\prod_{l=1}^{k}(T - \sigma_l I) = X(\Lambda_k R^{-1}\Lambda_k^{-1})(\Lambda_k L^{-1})$$

for $\sigma_k \to \lambda_1$, the largest eigenvalue of T, so the strict upper triangular elements of $\Lambda_k R^{-1}\Lambda_k^{-1}$ satisfy

$$r_{ij}(\prod_{l=1}^{k}(\lambda_i - \sigma_l)/\prod_{l=1}^{k}(\lambda_j - \sigma_l)) \to 0, (i < j, k \to \infty),$$

i.e.

$$\Lambda_k R^{-1}\Lambda_k^{-1} \to I. \tag{11}$$

Now doing a QL decomposition to $\Lambda_k R^{-1}\Lambda_k^{-1}$,

$$\Lambda_k R^{-1}\Lambda_k^{-1} = \tilde{Q}_k \tilde{L}_k \tag{12}$$

we see that $\tilde{L}_k$'s diagonal elements are all positive, and, according to (11)

$$\tilde{Q}_k \to I, \tilde{L}_k \to I$$

Denote the diagonal matrix

$$D_k = \mathrm{diag}[\mathrm{sign}((\tilde{L}_k\Lambda_k L^{-1})_{11}), \mathrm{sign}((\tilde{L}_k\Lambda_k L^{-1})_{22}), \ldots, \mathrm{sign}((\tilde{L}_k\Lambda_k L^{-1})_{nn})].$$

Then

$$\prod_{l=1}^{k}(T - \sigma_l I) = (X\tilde{Q}_k)(\tilde{L}_k\Lambda_k L^{-1}) = (X\tilde{Q}_k D_k)(D_k\tilde{L}_k\Lambda_k L^{-1}) \tag{13}$$

where $X\tilde{Q}_k D_k$ is an orthogonal and $D_k\tilde{L}_k\Lambda_k L^{-1}$ is a lower triangular with positive diagonal elements.

On the other hand, from QL iteration formula (7), it is easily seen that

$$\prod_{l=1}^{k}(T - \sigma_l I) = (Q_1 Q_2 \cdots Q_k)(L_k \cdots L_2 L_1) \tag{14}$$

By (13) and (14), it is further seen that such a QL decomposition is unique

$$Q_1 Q_2 \cdots Q_k = X\tilde{Q}_k D_k. \tag{15}$$

Also from (7) and using (10), (12) and (15)

$$\begin{aligned} T^{(k+1)} &= (Q_1 Q_2 \cdots Q_k)^T T (Q_1 Q_2 \cdots Q_k) \\ &= (X\tilde{Q}_k D_k)^T (X\Lambda X^{-1})(X\tilde{Q}_k D_k) \\ &= D_k\tilde{Q}_k^T\Lambda\tilde{Q}_k D_k \to \Lambda \end{aligned}$$

which is (9).

By Lemma 1

$$M[\mathbf{p}_1|\mathbf{p}_2|\cdots|\mathbf{p}_n] = [\mathbf{p}_1|\mathbf{p}_2|\cdots|\mathbf{p}_n]T^{(k+1)}$$

using Lemma 1 and Lemma 3, the results of this theorem are proved. □

Thus we can give a similar result to Algorithm 1 without proof as follows:

Theorem 2. *If $\sigma_k \to \lambda_1$, the largest generalized eigenvalue of problem* (1), *with rate p ($k \to \infty$) during executing Algorithm* 1, *then*

(i) $\mathbf{q}_1 \to \mathbf{z}_1$, $\mu_1 \to \lambda_1$, $b_1 \to 0$, *(with rate p).*

(ii) $(a_i, \mathbf{q}_i) \to (\lambda_i, \mathbf{z}_i)$, *for* $i = 2, 3, \ldots, m$.

4. Computational Aspects

4.1 Construction of $f(x)$ and selection of σ_1

As it turns out, any iterative function $f(x)$ and initial value σ_1 is suitable if it can guarantee that (4) holds during the iteration process (3).

Now we take Newton's iterative function as $f(x)$. Denoting the eigenpolynomial of matrix pencil (A, B) as

$$d(t) = \det[tB - A],$$

then (3) becomes

$$\sigma_{k+1} = \sigma_k - \frac{d(\sigma_k)}{d'(\sigma_k)} \tag{16}$$

and choose σ_1 as an upper bound of spectrum of (A, B), i.e. $\sigma_1 > \lambda_1$, (there are many methods for this, see [2]), then we can prove that:

Theorem 3. *If $\{\sigma_k\}$ is computed by* (16) *with $\sigma_1 > \lambda_1$, then $\{\sigma_k\}$ decreases monotonically and converges to λ_1 with a square rate of convergence.*

Proof. Let $p_j^{(n-1)}$ be the determinant of the order $n-1$ submatrix of $\sigma_1 B - A$ deleting the j-th row and column, let $p_{ij}^{(n-2)}$ be the determinant of the $(n-2)\times(n-2)$ submatrix of $\sigma_1 B - A$ deleting the i-th and j-th rows and columns. Obviously they are all greater than zero.

Since λ_1 is the largest eigenvalue of (1) and $\sigma_1 > \lambda_1$, there is no eigenvalue in (λ_1, ∞) and

$$d(\sigma) = \det[\sigma B - A] > 0, \quad \forall \sigma \in (\lambda_1, \infty).$$

while

$$d'(\sigma_1) = -(-1)^n \sum_{j=1}^{n} (-1)^{n-1} p_j^{(n-1)} > 0$$

and

$$d''(\sigma_1) = (-1)^n \sum_{j=1}^{n} \sum_{i=j+1}^{n} (-1)^{n-2} p_{ij}^{(n-2)} > 0.$$

Because all eigenvalues are strictly different, the roots of $d'(t)$ separate the ones of $d(t)$ and are separated by the ones of $d''(t)$ by Rolle's Theorem. Thus, $d'(t)$ and $d''(t)$ do not change their signs in $[\lambda_1, \infty)$.

Combining the above, we know that $d(t)$ and $d'(t)$ are both greater than zero and strictly monotonically increasing in (λ_1, ∞).

By induction, it is not hard to prove

$$\lambda_1 < \sigma_{k+1} < \sigma_k, (\forall k) \tag{17}$$

Inequality (17) means that $\{\sigma_k\}$ is a monotone and bounded sequence, so its limit exists. i.e.,

$$\lim_{k\to\infty} \sigma_k = \lambda_1$$

□

Remark 4.1. By the proof, we know the condition "$\sigma_k > \lambda_1, (\forall k)$" holds for the selected iteration function (16). As was pointed out in Remark 2.1 this will make the coefficient matrices in step 2 positive definite so that some better methods can be used to solve them.

Remark 4.2. By Lemma 3, as higher the rate of iterative function is, as faster the convergence of $(\lambda_1,\ \mathbf{z}_1)$ is. Thus some higher convergent rate function can be used to accelerate the speed of convergence. For example, with

$$f(t) = t - u(t)\frac{d(t-u(t))-d(t)}{2d(t-u(t))-d(t)}, \quad (u(t) = \frac{d(t)}{d'(t)})$$

one only needs to calculate two function values and one derivative value of $d(t)$, but its asymptotic convergence rate is four (see [18]).

4.2 Evaluation of $d(t)$ and $d'(t)$

Since there always occur some $d(t)$ and $d'(t)$ in the expression of $f(x)$, we consider now how to calculate them.

If both A and B are tridiagonal forms then we can calculate $d(t)$ and $d'(t)$ by a recursion which is similar to Hyman's methods (see [18] and [10]).

Let $A = \text{tridiag}[\beta_{i-1}, \alpha_i, \beta_i], B = \text{tridiag}[\delta_{i-1}, \gamma_i, \delta_i]$ and let

$$\begin{cases} \rho_0 = 1, \rho_1 = t\gamma_1 - \alpha_1 \\ \rho_i = (t\gamma_i - \alpha_i)\rho_{i-1} - (t\delta_{i-1} - \beta_{i-1})^2\rho_{i-2}, \quad i = 2, 3, \ldots, n \end{cases} \tag{18}$$

Then

$$\begin{cases} \rho_0' = 0, \rho_1' = \gamma_1 \\ \rho_i' = (t\gamma_i - \alpha_i)\rho_{i-1}' + \gamma_i\rho_{i-1} - (t\delta_{i-1} - \beta_{i-1})^2\rho_{i-2}' \quad -2(t\delta_{i-1} - \beta_{i-1})\delta_{i-1}\rho_{i-2}, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad i = 2, 3, \ldots, n \end{cases} \tag{19}$$

and

$$\rho_n = d(t), \quad \rho_n' = d'(t) \tag{20}$$

However, the scheme in (18), (19), (20) may be unstable for underflow or overflow, so we adjust it by scaling.

Let

$$\xi_i = \frac{\rho_i}{\rho_{i-1}}, \quad \eta_i = -\frac{\rho_i'}{\rho_i}, \quad i = 1, 2, \ldots, n$$

then

$$\begin{cases} \xi_1 = t\gamma_1 - \alpha_1 \\ \xi_i = (t\gamma_i - \alpha_i) - (t\delta_{i-1} - \beta_{i-1})^2/\xi_{i-1}, \quad i = 2, 3, \ldots, n \end{cases} \tag{21}$$

$$\begin{cases} \eta_0 = 0, \eta_1 = -\frac{\gamma_1}{\xi_1} \\ \eta_i = \frac{1}{\xi_i}\{(t\gamma_i - \alpha_i)\eta_{i-1} - \gamma_i - (t\delta_{i-1} - \beta_{i-1})[(t\delta_{i-1} \quad -\beta_{i-1})\eta_{i-2} - 2\delta_{i-1}]/\xi_{i-1}\}, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad i = 2, 3, \ldots, n \end{cases} \tag{22}$$

and we have

$$\frac{d(t)}{d'(t)} = -\eta_n^{-1}.$$

We take $\xi_1 = (t\delta_1 - \beta_1)^2\varepsilon^2$ when $\xi_1 = 0$ and $\xi_i = (t\delta_{i-1} - \beta_{i-1})^2\varepsilon/\xi_{i-1}$ when $\xi_i = 0$, $(i > 1)$ to prevent the breaking down in process, where ε means the machine precision. Then we can know that, by using a result in [10],

$$fl(\det[tB - A]) = \det[tB - A + E] * (1 + \tau)$$

where the matrix

$$E = \text{tridiag}[(t\delta_{i-1} - \beta_{i-1})\varepsilon_{i-1}, 0, (t\delta_i - \beta_i)\varepsilon_i]$$

and

$$|\varepsilon_i| \leq 2.5\varepsilon + O(\varepsilon^2),$$
$$|\tau| \leq (3n - 2)\varepsilon + O(\varepsilon^2).$$

If $\sigma_k B - A$ is not a tridiagonal, we can use a linear combination of $d(\sigma_k)$, $d(\sigma_{k-1})$, $\ldots$, $d(\sigma_{k-l+1})$ instead of $d'(\sigma_k)$. These function values could be computed when the linear systerms in step 2(a) are solved by LR (or incomplete LR) factorization.

For example, the simplest situation is $l = 2$, thus formula (16) becomes

$$\sigma_{k+1} = \sigma_k - \frac{d(\sigma_k)}{\frac{d(\sigma_k) - d(\sigma_{k-1})}{\sigma_k - \sigma_{k-1}}} \tag{23}$$

which is the secant line method, whose rate of convergent is 1.618.

It is easy to see that the extra computational complexity in calculating $d(t)$ or $d'(t)$ is $O(n)$ in both situations.

4.3 Solving linear equations

For an indefinite system as in step 4(a), there exists a way to partition the matrix into a 2×2 block form for which there are optimal methods to solve it for certain classes of problems, such as occuring in elliptic difference matrix problem (see [6], [13] and [20])

The systems in step 2(a) and 3 are symmetric positive definite, so the Algebraic Multi-Level Iteration method (AMLI) could be used. It produces a matrix M by recursion as a preconditoner of A, the coefficient matrix of a linear equations. Its main concept is as follows.

Let $A^{(l)} = A$ and $\Omega(N_k, S_k)$ be the matrix graph for the mathix $A^{(k)}$ of order n_k, $k = 0, 1, \ldots, l$ defined by a set of nodes N_k and a set of edges S_k where $a_{ij}^{(k)} \neq 0$ iff $(i, j) \in S_k$.

Assuming

$$N_0 \subset N_1 \subset \ldots \subset N_k \subset N_{k+1} \subset \ldots \subset N_l$$

and

$$\frac{n_{k+1}}{n_k} = \rho_k \geq \rho > 1$$

By permution and partition, the matrix $A^{(k+1)}$ has form

$$A^{(k+1)} = \begin{bmatrix} A_{11}^{(k+1)} & A_{12}^{(k+1)} \\ A_{21}^{(k+1)} & A_{22}^{(k+1)} \end{bmatrix} \begin{matrix} \} & (i,j) \in N_{k+1} \backslash N_k \\ \} & (i,j) \in N_k \end{matrix}$$

We take a sparse, symmetric positive definite matrix $B_{11}^{(k+1)}$ as an approximation of $(A_{11}^{(k+1)})^{-1}$, which satisfies

$$B_{11}^{(k+1)} A_{11}^{(k+1)} \mathbf{u}_1^{(k+1)} = \mathbf{u}_1^{(k+1)}$$

where

$$\mathbf{u}^{(k+1)} = \begin{bmatrix} \mathbf{u}_1^{(k+1)} \\ \mathbf{u}_2^{(k+1)} \end{bmatrix} \begin{matrix} \} & i \in N_{k+1} \backslash N_k \\ \} & i \in N_k \end{matrix}$$

is the restriction of a positive vector $\mathbf{u}^{(l)}$, for $A^{(l)}\mathbf{u}^{(l)} > 0$, to the nodes in N_{k+1}. Then we define $A^{(k)}$ as

$$A^{(k)} = A_{22}^{(k+1)} - A_{21}^{(k+1)} B_{11}^{(k+1)} A_{12}^{(k+1)}$$

It has been proved that $A^{(k)}$ is still a symmetric positive definite matrix for which holds that $A^{(k)}\mathbf{u}_2^{(k+1)} > 0$ (see [1]).

Now the preconditioning matrix sequence $\{M^{(k)}\}$ can be given as

$$\begin{cases} M^{(0)} = A^{(0)} \\ M^{(k+1)} = \begin{bmatrix} (B_{11}^{(k+1)})^{-1} & 0 \\ A_{21}^{(k+1)} & I \end{bmatrix} \begin{bmatrix} I & B_{11}^{(k+1)} A_{12}^{(k+1)} \\ 0 & \tilde{S}^{(k)} \end{bmatrix}, \quad k = 0, 1, \ldots, l-1 \end{cases}$$

where

$$\tilde{S}^{(k)} = A^{(k)}[I - P_{\nu_k}((M^{(k)})^{-1} A^{(k)})]^{-1}$$

and $P_{\nu_k}(x)$ is a polynomial with degree ν_k as well as $P_{\nu_k}(0) = 1$.

Thus, we get $M^{(l)}$, which is taken as the preconditioner of A.

The polynomial $P_{\nu_k}(x)$ can be chosen so that the level matrices $(M^{(k)})^{-1}A^{(k)}$ have very small condition numbers for all k. In fact, it holds that

$$\kappa[(M^{(l)})^{-1} A^{(l)}] = O(1), \quad (l \to \infty)$$

and the total ops is $O(n\log_2 n)$ in one iteration.The further discussions, see [1] and [4].

4.4 Short cycle strategy

In actual computations, we do not use the scheme defined by Algorithm 1 straightforwardly for two reasons:

1. The results in Theorem 1 takes place asymptotically when $k \to \infty$, but the actual value of k is not very large, usually when σ_{k+1} and σ_k are closed enough. So in fact we can always get a few m_0 $(a_i, \mathbf{q}_i)$s which are good approximates of $(\lambda_i, \mathbf{z}_i)$s. Obviously, the best strategy is to make these m_0 pairs accurate enough first, then from them to get new rough eigenpairs.

2. Step 3 is a Lanczos-like procedure. It is well known that it would produce surplus copies when the number of iterations is a bit large. So, we could avoid this situation by using only few Lanczos steps in every processing.

We always take $m_0 \leq 5$ in the computation, then check and abandon some copies of what we have got – which is not difficult to judge when the distance of two eigenvalues as well as the angle between their corresponding eigenvector are calculated (see example 3 in Section 5). Next we rerun this procedure from the newest eigenvalue and repeat the process until m eigenpairs are calculated fully.

Executing the scheme from $j = 2$ is equivalent to the situation that deleting the first row and column of matrix $T^{(k+1)}$ and going on with the shifted QL process continously when α_1 converges. So it is not hard to prove the similar results as in Theorem 2.

Now we rewrite the algorithm as follows:

Algorithm NLRI. Choose σ_1 and $\mathbf{u}_1$, ($\|\mathbf{u}_1\|_B = 1$).
$k = 1$, $K_0 =$ a large natural number
WHILE $k < K_0$
  Solve $(\sigma_k B - A)\tilde{\mathbf{u}}_{k+1} = -B\mathbf{u}_k$
  $\mathbf{u}_{k+1} = \tilde{\mathbf{u}}_{k+1}/\|\tilde{\mathbf{u}}_{k+1}\|_B$, $\sigma_{k+1} = \sigma_k - d(\sigma_k)/d'(\sigma_k)$
  IF $\|\tilde{\mathbf{u}}_{k+1}\|_B^{-1} > \varepsilon_1$ AND $|\sigma_{k+1} - \sigma_k| > \varepsilon_2|\sigma_{k+1}|$ THEN
    $k = k + 1$
  ELSE
    $\mathbf{v}_1 = \mathbf{u}_{k+1}$, $\mu_1 = (A\mathbf{v}_1, \mathbf{v}_1)$, $K_0 = k$
  END
END
$j = 1$, $l = 1$,
WHILE $l < m$
  $a_j = \mu_j$, $b_{j-1} = 0$, $\mathbf{q}_j = \mathbf{v}_j$, m_0
  FOR $i = j, \ldots, m_0$
    Solve : $B\mathbf{r} = A\mathbf{q}_i$
    $\mathbf{r} = \mathbf{r} - a_i\mathbf{q}_i - b_{i-1}\mathbf{q}_{i-1}$
    $b_i = \sqrt{(B\mathbf{r}, \mathbf{r})}$, $\mathbf{q}_{i+1} = \mathbf{r}/b_i$, $a_{i+1} = (A\mathbf{q}_{i+1}, \mathbf{q}_{i+1})$
  END
  FOR $i = j + 1, \ldots, m_0 + 1$
    $\sigma_1 = a_i$, $\mathbf{u}_1 = \mathbf{q}_i$.
    $k = 1$, $K_0 =$ a large natural number
    WHILE $k < K_0$
      Solve : $(\sigma_k B - A)\tilde{\mathbf{u}}_{k+1} = -B\mathbf{u}_k$
      $\mathbf{u}_{k+1} = \tilde{\mathbf{u}}_{k+1}/\|\tilde{\mathbf{u}}_{k+1}\|_B$, $\sigma_{k+1} = (A\mathbf{u}_{k+1}, \mathbf{u}_{k+1})$
      IF $\|\tilde{\mathbf{u}}_{k+1}\|_B^{-1} > \varepsilon_1$ AND $|\sigma_{k+1} - \sigma_k| > \varepsilon_2|\sigma_{k+1}|$ THEN
        $k = k + 1$
      ELSE
        $\mathbf{v}_i = \mathbf{u}_{k+1}$, $\mu_i = \sigma_{k+1}$, $K_0 = k$
      END
    END
  END
  Check μ_{j+1}, μ_{j+2}, $\ldots$, μ_{m_0+1}
  Mark surplus copies
  Decide the initial vector μ_{j+j_0} for next cycle
  $j = j + j_0$
  $l = l + m_0 -$ number of copies
END

4.5 Parallel execution

When $(a_2, \mathbf{q}_2)$, $(a_3, \mathbf{q}_3)$, ..., $(a_{m_0+1}, \mathbf{q}_{m_0+1})$, the approximations of the m_0 largest eigenpairs, are computed after step (3), each of them can be worked out in parallel in step (4) for getting accurate eigenpairs $(\lambda_2, \mathbf{z}_2)$, $(\lambda_3, \mathbf{z}_3)$, ..., $(\lambda_{m_0+1}, \mathbf{z}_{m_0+1})$. In other words, it is fully parallelizable with respect to the calculations done for each $(\lambda_i, \mathbf{z}_i)$.

Step (3) is actually a Lanczos process. Except solving linear equations with matrix B, the remaining part of the computation consists of matrix-vector multiplications which are parallelizable as long as only nearest neighbours are needed, which holds for regular finite element matrices.

Solving symmetric positive definite systems in step 2(a) and 3 with AMLI are also parallelizable, the details can be found in [5].

5. Numerical Examples

All examples below were computed on SUN SPARCserver 10 Model 514Mp computer in Department of Mathematics, University of Nijmegen. The "exact eigenpairs" stands for those computed by MATLAB.

Example 1. We take A as the matrix representation of the second difference operator which is a tridiagonal matrix with entries 1, -2, 1, respectively on the super-diagonals and subdiagonals, i.e.

$$A = \text{tridiag}[1, -2, 1], \qquad B = I.$$

The order of matrix n is taken as 128, 256, 400 respectively and $\varepsilon_2 = 10^{-11}$. The results are shown in Tables 1 and 2 - where m^* denotes the number of eigenpairs we get (including copies) when m true extreme eigenvalues converge. The meanings of $(\lambda_i, \mathbf{z}_i)$ and $(a_i, \mathbf{q}_i)$ are as before and $(\mu_{j(i)}, \mathbf{v}_{j(i)})$ is the calculated eigenpair corresponding to $(\lambda_i, \mathbf{z}_i)$. $\theta(\mathbf{q}_i, \mathbf{z}_i) = \cos^{-1}((B\mathbf{q}_i, \mathbf{z}_i))$ means "angle" between $\mathbf{q}_i$ and $\mathbf{z}_i$. " Ite." is the number of using RQI.

We set $j_0 = m_0 = 1$ always as long as μ_{j+1} is not a copy of some μ_i $(i < j)$. "(k)" in column 2 means that the kth converging eige.

From Table 1, we see that $(a_i,\ \mathbf{q}_i)$ is really a very good approximation of $(\lambda_i,\ \mathbf{z}_i)$ for almost all situations. So it is reasonable to take it as an initial approximation in the RQI process.

Table 1. Solving 15 Largest (Smallest) Eigenpairs of A_{256}

Exact Eigenvalue	Order of Convergence	$\lvert\lambda_i - \mu_{j(i)}\rvert$	$\lvert a_i - \lambda_i\rvert$	$\theta(\mathbf{q}_i, \mathbf{z}_i)$	Ite.
-0.00014942666053	1 (9)	-	-	-	-
-0.00059768431381	2	7.44×10^{-16}	8.94×10^{-6}	0.11	2
-0.00134470597819	3 (11)	7.12×10^{-16}	9.30×10^{-6}	0.09	2
-0.00239038002871	4	5.46×10^{-16}	8.94×10^{-6}	0.08	2
-0.00373455021380	5 (12)	8.20×10^{-16}	9.67×10^{-6}	0.07	2
-0.00537701567859	6	2.79×10^{-16}	1.34×10^{-5}	0.06	2
-0.00731753099495	7	3.82×10^{-16}	6.71×10^{-5}	0.12	2
-0.00955580619817	8	1.24×10^{-13}	2.68×10^{-3}	0.34	3
-0.01209150683024	10	2.42×10^{-13}	6.81×10^{-4}	0.25	2
-0.01492425398990	15	7.95×10^{-14}	5.65×10^{-4}	0.24	2
-0.01805362438919	14	6.94×10^{-17}	5.67×10^{-5}	0.30	2
-0.02147915041675	13	2.67×10^{-16}	9.88×10^{-4}	1.81	4
-0.02520032020766	16	3.89×10^{-16}	2.30×10^{-5}	0.03	2
-0.02921657771995	17	1.04×10^{-17}	1.01×10^{-1}	1.18	4
-0.03352732281766	18	2.82×10^{-15}	4.08×10^{-4}	0.19	2

-3.99985057333947	1	-	-	-	-
-3.99940231568619	2	4.44×10^{-15}	8.94×10^{-6}	0.11	2
-3.99865529402181	3 (6)	1.02×10^{-13}	9.40×10^{-6}	0.09	2
-3.99760961997129	4 (8)	4.00×10^{-15}	8.98×10^{-6}	0.08	2
-3.99626544978620	5	1.52×10^{-12}	2.08×10^{-4}	0.31	2
-3.99462298432141	7	6.66×10^{-15}	4.94×10^{-5}	0.35	2
-3.99268246900505	9	1.51×10^{-13}	4.68×10^{-3}	0.33	5
-3.99044419380183	10	1.33×10^{-15}	4.30×10^{-5}	0.13	2
-3.98790849316976	11 (12)	1.24×10^{-14}	3.22×10^{-6}	0.02	2
-3.98507574601010	13 (16)	4.00×10^{-15}	1.95×10^{-5}	0.06	2
-3.98194637561081	14 (15)	1.20×10^{-14}	8.58×10^{-4}	0.33	3
-3.97852084958325	17	2.66×10^{-15}	6.21×10^{-3}	0.47	5
-3.97479967979234	18	1.58×10^{-12}	2.78×10^{-6}	0.03	1
-3.97078342228005	19 (20)	3.55×10^{-15}	9.68×10^{-4}	0.26	3
-3.96647267718234	21	1.33×10^{-15}	8.42×10^{-4}	0.54	3

Table 2. Numerical Results about A_n

n	Solving Largest Eigenvalues			Solving Smallest Eigenvalues		
	m	m^*	$\max \lvert\lambda_i - \mu_{j(i)}\rvert$	m	m^*	$\max \lvert\lambda_i - \mu_{j(i)}\rvert$
128	7	7	1.38×10^{-14}	7	8	1.20×10^{-14}
256	15	18	2.42×10^{-13}	15	21	1.52×10^{-12}
400	10	11	3.53×10^{-13}	5	5	1.47×10^{-14}

Example 2. Consider the differential equation eigenvalue problem[3]

$$\begin{cases} u''(x) + \lambda u(x) = 0, & x \in (0,1) \\ u'(0) = 0, u(1) = 1 \end{cases}$$

and the corresponding algebraic eigenvalue problem

$$A_h \mathbf{u}_h = \lambda B_h \mathbf{u}_h$$

where $A_h = \frac{1}{h}(\text{tridiag}[-1, 2, -1] - \mathbf{e}_1\mathbf{e}_1^T)$, $B_h = \frac{h}{3}(\text{tridiag}[0.5, 2, 0.5] - \mathbf{e}_1\mathbf{e}_1^T)$.

We take $h = \frac{1}{16}$, $\frac{1}{64}$ and $\frac{1}{256}$ respectively and solve its several largest and smallest eigenvalues as well as corresponding eigenvectors. The main results are found in Table 3.

Table 3. Numerical Results for (A_h, B_h)

$\frac{1}{h}$	Solving Largest Eigenvalues			Solving Smallest Eigenvalues		
	m	m^*	$\max \frac{\lvert\lambda_i - \mu_{j(i)}\rvert}{\lvert\lambda_i\rvert}$	m	m^*	$\max \frac{\lvert\lambda_i - \mu_{j(i)}\rvert}{\lvert\lambda_i\rvert}$
16	6	7	1.58×10^{-12}	6	7	5.04×10^{-14}
64	10	13	5.88×10^{-11}	10	12	1.52×10^{-12}
256	10	15	6.10×10^{-12}	12	14	7.81×10^{-10}

Example 3. We take Wilkinson's matrix W_{21}^+ as A and identity matrix as B, i.e. it is a standard eigenvalue problem

$$W_{21}^+ \mathbf{x} = \lambda \mathbf{x}$$

We compute its 8 largest eigenvalues and corresponding eigenvectors because λ_{2i-1} and λ_{2i} (i=1,2,3,4) are so close that we can treat them as a pair of multiple roots in a numerical computational sense, especially, the distance of λ_1 and λ_2 is around $O(10^{-14})$ as is well known. We notice that for such a matrix like W_{21}^+, sometimes we could only find one eigenvalue from two very close ones. The situations showed as Table 4.

Table 4. Convergence Order of 10 Largest Eigenvalues of W_{21}^{+}

Exact Eigenvalue	$\varepsilon_2 = 10^{-13}$			$\varepsilon_2 = 10^{-12}$	
	$j_0 = m_0 = 2$	$j_0 = m_0 = 3$	varied $j_0^{(a)}$	varied $j_0^{(a)}$	varied $j_0^{(b)}$
10.74619418290339	1	1	1 (3)	1	1
10.74619418290332	2	2	2	2	2
9.21067864736133	3 (4)	3	4	3 (5)	3
9.21067864730492	not found	not found	5 (6)	4	4
8.03894112282902	5 (6)	8	10	8	5 (6)
8.03894111581427	not found	7	11	9	not found
7.00395220952868	7	not found	7	7	7
7.00395179861637	8	4 (5)	8	6	8

Varied $j_0^{(a)}$: $j_0 = 1$ always unless μ_{j+1} is a copy or there exists a $\mu_{j+i} > \mu_{j+1}$, $(1 < i \leq m_0)$ which is not a copy of any other ones.

Varied $j_0^{(b)}$: $j_0 = m_0$ always unless μ_{j+m_0} is a copy or it looks like that there exists a gap between μ_{j+1} and μ_{j+m_0}.

In addition, we find that, for example as column 2, when μ_2 converges it is very close to μ_1 and we must ask : is it a copy of μ_1? The answer is no because the angle between $\mathbf{z}_2$ and $\mathbf{z}_1$ is nearly vertical ($\approx 0.44\pi$). Later, μ_4 is also close to μ_3 but this time we can be sure it is a copy of μ_3 for the angle of their eigenvectors is almost zero ($\approx 0.03\pi$). (Same for other close pairs.)

6. Concluding Remarks

It has been shown that it can be advantageous to use the algorithm presented in this paper to compute some extreme eigenpairs of a generalized (or standard, of course) large scale sparse eigenvalue problem because of its simplicity of getting initial vectors which suffices to make the iterative procedure converge to the expected eigenpairs, preserving its sparsity structure, its use of algebraic multi-level method and its parallelization. This method leads to a better result than precious computations even for such a matrix as W_{21}^{+}.

In actual calculation, there is no need to solve the linear systems in step 2(a) and produce every vector $\mathbf{u}_k$ in each iterative loop from the beginning. A more efficient method is that we only use the function $f(x)$ to get new σ_{k+1} first until $|\sigma_{k+1} - \sigma_k|$ has become very small and then execute the whole step 2 from this σ_{k+1} (which is equivalent to executing the algorithm from a more accurate initial σ_1).

Taking too small criterion ε_2 in step 2 may lead to a larger error during the Lanczos-like procedure in step 3 due to dividing by b_1 ($b_1 = \beta_1^{(k+1)}$ as proved in Lemma 1 and $\beta_1^{(k+1)}$ is less than ε_2 when the equivalent QL process (7) is convergent, see [11]). A reasonable strategy is taking suitable ε_2 (as $10^{-5} \sim 10^{-10}$, for example) at first to get computable results and then do RQI for each pair with smaller ε_2 aga in to obtain a more accurate solution.

References

[1] Axelsson, O., The method of diagonal compensation of reduced matrix entries and multilevel iteration, *J. Comput. Appl. Math.*, 38, 31-43, 1991.

[2] Axelsson, O., *Iterative Solution Methods*, Cambridge University Press, 1994.

[3] Axelsson, O. and Neytcheva, M., Finding eigenvalues in an interval using parallelizable algorithm, *Proceedings of the Workshop on Parallel Algorithms (WPA'92)*, Ed. I. Dimov and O. Tonev, Bankja, Bulgaria, 11-21, 1992.

[4] Axelsson, O. and Neytcheva, M., Algebraic multilevel iteration method for Stieltjes matrices, *J. Num. Lin. Alg. Appl.*, 1(3), 213-236, 1994.

[5] Axelsson, O. and Neytcheva, M., Parallel implementations of the algebraic multilevel iteration method, *Proceedings of the fifth SIAM conference on Applied Linear Algebra*, Ed. J.G.Lewis, SIAM,372-376.

[6] Axelsson, O. and Vassilevsky, P., Construction of variable-step preconditioners for inner-outer iteration methods,*Iterative Methods in Linear Algebra*, Ed. R.Beauwens and P.de Groen, North Holland, 1-14, 1992.

[7] Davison, E.R., The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices, *J. Comput. Physics*, 17, 87-94, 1975.

[8] Jiang Erxiong, On the convergence of diagonal element and asymptotic convergence rate for the shifted tridiagonal QL algorithm, *J. Comp. Math.*, 3(3), 252-261, 1985.

[9] Jiang Erxiong, An algorithm for finding generalized eigenpairs of a symmetric definite matrix pencil, *Linear Algebra Appl.*, 132,65-92, 1990.

[10] Li, T.Y. and Zeng, Z., The Laguerre iteration in solving the symmetric tridiagonal eigenproblem, *SIAM J. Sci. Comput.* , 15(5), 1145-1173, 1994.

[11] Parlett, B.N., *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, N.J., 1980.

[12] Reinsch, C. and Bauer, F.L., Rational QR transformation with Newton shift for symmetric tridiagonal matrices, *Num. Math.*, 11, 264-272, 1968.

[13] Vassilevsky, P., Preconditioning nonsymmetric and indefinite finite element elliptic matrices, *J. Num. Lin. Alg. Appl.*, 1(1), 59-76, 1992.

[14] Wilkinson, J.H., *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

[15] Xu Jinchao, A new class of iterative methods for nonselfadjoint or indefinite problems, *SIAM J.Numer.Anal.*, 29(2), 303-319.

[16] Yu Chonghua, *On symmetric QL algorithm*, Doctoral Dissertation, Fudan University, 1987.

[17] Yu Chonghua, Convergence of eigenvalues in the natural Order for the QL algorithm applied to symmetric tridiagonal matrices, *Numer. Math. J. Chinese Univ.*, 10(1), 28-40, 1988.

[18] Yu Chonghua, On accelerating a symmetric QL algorithm by using compound shift, *Numer. Math. J. Chinese Univ.*, 13(3), 263-272, 1991.

[19] Yu Chonghua, A convergence condition for diagonal elements in the shifted symmetric QL algorithm, *Numer. Math. J. Chinese Univ.*, 15(1), 22-31, 1993.

[20] Yserentant, H., Preconditioning indefinite discretization matrices, *Numer. Math.*, 54,719-734, 1988.