MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

Iterative Refinement for Symmetric Eigenvalue Decomposition Adaptively Using Higher-Precision Arithmetic

Takeshi OGITA and Kensuke AISHIMA

METR 2016–11 June 2016

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.keisu.t.u-tokyo.ac.jp/research/techrep/index.html

Iterative Refinement for Symmetric Eigenvalue Decomposition Adaptively Using Higher-Precision Arithmetic

Takeshi OGITA
Division of Mathematical Sciences
School of Arts and Sciences
Tokyo Woman's Christian University
ogita@lab.twcu.ac.jp

Kensuke AISHIMA
Department of Mathematical Informatics
Graduate School of Information Science and Technology
The University of Tokyo
Kensuke_Aishima@mist.i.u-tokyo.ac.jp

June 2016

Abstract

Efficient refinement algorithms are proposed for symmetric eigenvalue problems. The structure of the proposed algorithms is straightforward, primarily comprising matrix multiplications. We first present a basic algorithm for improving all the eigenvectors associated with well-separated eigenvalues. We show that it quadratically converges in exact arithmetic, provided that a modestly accurate initial guess is given. The convergence rate is also preserved in finite precision arithmetic if the working precision is sufficiently high in the algorithm, i.e., it is indispensable to double the working precision in each iteration. Moreover, for multiple eigenvalues, we prove quadratic convergence under a technical assumption, whenever all the simple eigenvalues are well separated. We emphasize that this approach to multiple eigenvalues overcomes the limitation of our analysis to real matrices, resulting in the extension of the proof to Hermitian matrices. On the basis of the basic algorithm, we propose the complete version of a refinement algorithm which can also improve the eigenvectors associated with clustered eigenvalues. The proposed algorithms construct an accurate eigenvalue decomposition of a real symmetric matrix by iteration, up to the limit of computational precision. Numerical results demonstrate excellent performance of the proposed algorithms in terms of convergence rate and overall computational cost, and show that the basic algorithm is considerably faster than a naive approach using multiple-precision arithmetic.

1 Introduction

Let A be a real symmetric $n \times n$ matrix. We are concerned with the symmetric eigenvalue problem $Ax = \lambda x$, where $\lambda \in \mathbb{R}$ is an eigenvalue of A and $x \in \mathbb{R}^n$ is an eigenvector of A associated with λ. Solving this problem is important because it plays a significant role in scientific computing. Excellent overviews can be found in [22, 26].

Throughout the paper, I and O denote the identity and the zero matrices of appropriate size, respectively. Moreover, $\|\cdot\|$ denotes the Euclidean norm for vectors and the spectral norm for matrices. For legibility, if necessary, we distinguish between the approximate quantities and the computed results, e.g., for some quantity α we write $\widetilde{\alpha}$ and $\widehat{\alpha}$ as an approximation of α, and a computed result for α, respectively. The relative rounding error unit according to ordinary floating-point arithmetic is denoted by $\mathbf{u}$. For example, $\mathbf{u} = 2^{-53}$ for IEEE 754 binary64.

For simplicity, we basically handle only real matrices. The discussions in this paper can be extended to Hermitian matrices, as we will mention at Subsection 4.3. Moreover, the discussion for the symmetric eigenvalue problem can readily be extended to the generalized symmetric (or Hermitian) definite eigenvalue problem $Ax = \lambda Bx$ where A and B are real symmetric (or Hermitian) with B being positive definite.

1.1 Our purpose

This paper aims to develop an algorithm for calculating an arbitrarily accurate result of the eigenvalue decomposition

$$A = XDX^{\mathrm{T}}, \tag{1}$$

where X is the $n \times n$ orthogonal matrix whose ith columns are the eigenvectors $x_{(i)}$ of A (called the eigenvector matrix), and D is the $n \times n$ diagonal matrix whose diagonal elements are the corresponding eigenvalues $\lambda_i \in \mathbb{R}$, i.e., $D_{ii} = \lambda_i$ for $i = 1, \ldots, n$. For this purpose we discuss iterative refinement methods for (1) together with the convergence analysis.

Several efficient numerical algorithms for (1) have been developed such as the bisection method with inverse iteration, the QR algorithm, the divide-and-conquer algorithm or the Multiple Relatively Robust Representations (MRRR) algorithm via Householder's tridiagonalization, and the Jacobi algorithm. For details, see, [9, 10, 13, 14, 22, 26] and references cited therein. Since they have actively been studied in numerical linear algebra for decades, there are highly reliable implementations for them, such as Linear Algebra Package (LAPACK) routines.

We stress that we do not intend to compete with such existing algorithms but to develop a refinement algorithm for improving the results obtained by any of them. Such an algorithm is useful if the quality of the results are not satisfactory. Namely, the proposed algorithm can be regarded as a supplement to the existing ones for constructing (1). In fact, we assume that some computed result $\widehat{X}$ for (1) can be obtained by backward stable algorithms in ordinary floating-point arithmetic. Our analysis provides a sufficient condition for the convergence of the iterations.

In our proposed algorithms, the use of higher-precision arithmetic is mandatory, but is basically restricted to matrix multiplication, which accounts for most of the computational cost. For example, an approach used in Extra Precise Basic Linear Algebra Subroutines (XBLAS) [18] and other accurate and efficient algorithms for dot products [20, 23, 24] and matrix products [21] based on error-free transformations are available for practical implementation.

1.2 Background

A naive and possible approach to achieving an accurate eigenvalue decomposition is to use a multiple-precision arithmetic library such as MPFR [19] with GMP [12] in Householder's tridiagonalization and the subsequent algorithm. In general, however, we do not know in advance how much arithmetic precision suffices to achieve the desired accuracy for results. Moreover, the use of such multiple-precision arithmetic for entire computations is often much more time-consuming than ordinary floating-point arithmetic, owing to the difficulty of optimization for today's computer architectures. Therefore, we prefer the approach of the iterative refinement, rather than that of simply using multiple-precision arithmetic.

There exist several refinement algorithms for eigenvalue problems that are based on Newton's method (cf. e.g., [3, 5, 11, 25]). Since this sort of algorithm is designed to improve eigenpairs $(\lambda, x) \in \mathbb{R} \times \mathbb{R}^n$ individually, applying such a method to all eigenpairs requires $\mathcal{O}(n^4)$ arithmetic operations. To reduce the computational cost, one may consider the preconditioning by Householder's tridiagonalization of A by using ordinary floating-point arithmetic such as $T \approx \widehat{H}^{\mathrm{T}} A \widehat{H}$, where T is a tridiagonal matrix, and $\widehat{H}$ is an approximately orthogonal matrix involving rounding errors. However, it is not a similarity transformation, so that the original problem is slightly perturbed. Therefore, the accuracy of eigenpairs is limited by the orthogonality of $\widehat{H}$.

Simultaneous iteration or Grassmann-Rayleigh quotient iteration in [1] can potentially be used to refine eigenvalue decompositions. However, such methods require the use of higher-precision arithmetic concerning the orthogonalization of the approximate eigenvectors, and hence we cannot restrict the higher-precision arithmetic to matrix multiplication. Moreover, Wilkinson [26, Chapter 9, pp. 637–647] explained the refinement of eigenvalue decompositions for general square matrices, mentioning Jahn's method [17, 4]. Such methods rely on a similarity transformation $C := \widehat{X}^{-1}A\widehat{X}$ with high accuracy for a computed result $\widehat{X}$ for X, which requires an accurate solution of the linear system $\widehat{X}C = A\widehat{X}$ for C, and breaks the symmetry of A.

Alternatively, the Jacobi algorithm is useful for improving the accuracy of all computed eigenvectors. In addition, Davies and Modi [6] proposed a direct method for completing the symmetric eigenvalue decomposition of nearly diagonal matrices. However, in fact, owing to rounding errors in floating-point arithmetic, it is difficult to compute the eigenvectors corresponding to clustered eigenvalues with high accuracy. In other words, higher-precision arithmetic is required for computing accurate eigenvectors corresponding to the clustered eigenvalues. We will mention the details in Section 2.

With such a background of the study, we try to derive a simple and efficient iterative refinement algorithm for simultaneously improving the accuracy of all the eigenvectors with quadratic convergence, which requires $\mathcal{O}(n^3)$ operations for each iteration. In fact, the proposed algorithm can be regarded as a variant of Newton's method, and therefore, its quadratic convergence is naturally derived.

1.3 Our idea

The idea to design the proposed algorithm is as follows. For a computed eigenvector matrix $\widehat{X}$ for (1), define $E \in \mathbb{R}^{n\times n}$ such that $X = \widehat{X}(I + E)$. Then we aim to compute a sufficiently precise approximation $\widetilde{E}$ of E using the following two relations:

$$\begin{cases} X^{\mathrm{T}}X = I & \text{(orthogonality)} \\ X^{\mathrm{T}}AX = D & \text{(diagonality)} \end{cases} \tag{2}$$

After obtaining $\widetilde{E}$, we can update $X' := \widehat{X}(I + \widetilde{E})$. If necessary, we iterate the process such as $\widehat{X}^{(\nu+1)} := \widehat{X}^{(\nu)}(I + \widetilde{E}^{(\nu)})$. Under some conditions, we prove $\widetilde{E}^{(\nu)} \to O$ and $\widehat{X}^{(\nu)} \to X$, where the convergence rates are quadratic.

Using (2), we will concretely derive a basic refinement algorithm (Algorithm 1 in Section 3), which works perfectly for the eigenvectors associated with well-separated eigenvalues. However, we encounter the problem on the convergence if A has clustered eigenvalues, since the convergence rate strongly depends on the relative gap of eigenvalues as will be stated in our convergence analysis (Theorem 1 in Section 4). In such cases, we need special care, especially for the eigenvectors associated with nearly multiple eigenvalues.

In general, it is notoriously difficult to obtain accurate approximation of the eigenvectors corresponding to clustered eigenvalues as stated by the

Davis–Kahan theorem for eigenpairs (cf. e.g., [22, Theorem 11.7.1]): Let $(\lambda, x) \in \mathbb{R} \times \mathbb{R}^n$ be some eigenpair of A with $\|x\| = 1$. For any $\widehat{x} \in \mathbb{R}^n$ with $\|\widehat{x}\| = 1$ and any $\mu \in \mathbb{R}$ whose closest eigenvalue is λ,

$$|\sin \angle(\widehat{x}, x)| \leq \frac{\|A\widehat{x} - \mu\widehat{x}\|}{gap(\mu)}, \quad gap(\mu) := \min\{|\mu - \lambda_i| : \lambda_i \neq \lambda\}. \tag{3}$$

However, the eigenspace spanned by all the eigenvectors corresponding to clustered eigenvalues is not sensitive to perturbations, which is also proved by Davis and Kahan (cf. e.g., [22, Theorem 11.7.2]). This theorem is crucial. Let $\mathcal{J}$ denote a set of the indexes of the eigenvalues in a cluster with $p := |\mathcal{J}|$, and $\mathcal{K} := \{1, 2, \dots, n\} \setminus \mathcal{J}$. Let $X_{\mathcal{J}}, \widehat{X}_{\mathcal{J}} \in \mathbb{R}^{n \times p}$ denote the matrices comprising all $x_{(j)}, \widehat{x}_{(j)}$ for $j \in \mathcal{J}$, respectively. In addition, let $\mathcal{X}_{\mathcal{J}}, \widehat{\mathcal{X}}_{\mathcal{J}}$ denote the subspaces spanned by the columns of $X_{\mathcal{J}}, \widehat{X}_{\mathcal{J}}$, respectively. Moreover, let $\{z_j\}_{j \in \mathcal{J}}$ be an orthonormal basis of $\widehat{\mathcal{X}}_{\mathcal{J}}$, and $Z_{\mathcal{J}} \in \mathbb{R}^{n \times p}$ the matrix that consists of $\{z_j\}_{j \in \mathcal{J}}$. Then, the Davis–Kahan theorem for subspaces states

$$\sin\left(\max_{\widehat{x} \in \widehat{\mathcal{X}}_{\mathcal{J}}} \min_{x \in \mathcal{X}_{\mathcal{J}}} |\angle(\widehat{x}, x)|\right) \leq \frac{\|AZ_{\mathcal{J}} - Z_{\mathcal{J}}S_{\mathcal{J}}\|}{gap_{\mathcal{J}}}, \ S_{\mathcal{J}} := Z_{\mathcal{J}}^{\mathrm{T}}AZ_{\mathcal{J}} \in \mathbb{R}^{p \times p}, \tag{4}$$

where $gap_{\mathcal{J}} := \min\{|\theta_i - \lambda_k| : 1 \leq i \leq p, \ k \in \mathcal{K}\}$ with θ_i being the eigenvalues of $S_{\mathcal{J}}$. In general, using floating-point arithmetic, $\widehat{X}, \widehat{D}$ obtained by backward stable algorithms satisfy

$$\|A\widehat{X} - \widehat{X}\widehat{D}\| = \mathcal{O}(\|A\|\mathbf{u}), \quad \|\widehat{X}^{\mathrm{T}}\widehat{X} - I\| = \mathcal{O}(\mathbf{u}), \tag{5}$$

where $\mathbf{u}$ is the relative rounding error unit. Since Weyl's perturbation theorem (cf. e.g., [9, Theorem 5.1]) indicates $|\lambda_i - \widehat{D}_{ii}| = \mathcal{O}(\|A\|\mathbf{u})$ for $i = 1, \dots, n$, the clustered eigenvalues are easily identified. In addition, the right-hand side of (4) is sufficiently small whenever $\|A\|/gap_{\mathcal{J}}$ is a constant of modest size.

From this viewpoint, we introduce some criterion for judging whether the eigenvalues are clustered, which is done in the basic refinement algorithm (Algorithm 1). For simplicity, let us consider the case where there is only one cluster of eigenvalues $\lambda_1 \approx \lambda_2 \approx \cdots \approx \lambda_p$, and the rest of the eigenvalues are well separated each other and from the cluster with $\max|\lambda_i|/\min|\lambda_i| \approx 1$. Then, $\mathcal{J} = \{1, 2, \dots, p\}$ and $\mathcal{K} = \{p+1, p+2, \dots, n\}$. Define α, β as

$$\alpha := \frac{\|A\|}{gap_{\mathcal{J}}}, \quad \beta := \frac{\|A\|}{\min_k gap(\lambda_k)} = \frac{\|A\|}{\min_{\lambda_i \neq \lambda_j} |\lambda_i - \lambda_j|}. \tag{6}$$

Note that $\alpha \approx 1$, and due to the Davis–Kahan theorems as (3) and (4),

$$\|\widehat{X}_{\mathcal{K}} - X_{\mathcal{K}}\| = \mathcal{O}(\alpha\mathbf{u}) = \mathcal{O}(\mathbf{u}), \quad \|\widehat{X}_{\mathcal{J}} - X_{\mathcal{J}}\| = \mathcal{O}(\beta\mathbf{u}).$$

In other words, $\widehat{X}_{\mathcal{J}}$ is not sufficiently accurate. However, if Algorithm 1 is performed in sufficiently high precision arithmetic, it transforms $\widehat{X} =$

$[\widehat{X}_{\mathcal{K}}, \widehat{X}_{\mathcal{J}}]$ to $X' = [X'_{\mathcal{K}}, X'_{\mathcal{J}}]$, where $X'_{\mathcal{K}}$ and $X'_{\mathcal{J}}$ correspond to the well-separated eigenvalues and the clustered ones, respectively, with

$$\|X'_{\mathcal{K}} - X_{\mathcal{K}}\| = \mathcal{O}(\mathbf{u}^2), \quad \|X'^{\mathrm{T}}X' - I\| = \mathcal{O}(\mathbf{u}^2).$$

The estimation of $X'_{\mathcal{K}}$ above is convincing because Algorithm 1 is viewed as a sort of Newton's method. See (40) and (41) in Lemma 1 for details. Regarding the orthogonality of X', since Algorithm 1 has an aspect of the Newton–Schulz iteration [16, Section 8.3] as shown in Remark 2, the quadratic convergence of the orthogonality is naturally derived. In addition, note that the estimation above is consistent with the numerical result for $\rho = 10^3$ in Section 5. Thus, we can see that the columns of $X'_{\mathcal{J}}$ are sufficiently accurate approximation of the orthonormal basis vectors of the eigenspace spanned by the columns of $X_{\mathcal{J}}$, even though $X'_{\mathcal{J}}$ is not necessarily close to $X_{\mathcal{J}}$. For this matrix $X'_{\mathcal{J}}$, we introduce the diagonal shift to the matrix $(X'_{\mathcal{J}})^{\mathrm{T}}AX'_{\mathcal{J}}$, which is the submatrix of $(X')^{\mathrm{T}}AX'$ corresponding to the clustered eigenvalues, to make the eigenvalues well-separated in the same manner as the MRRR algorithm [10] and others. After such a preconditioning, we can obtain an accurate approximation $X''_{\mathcal{J}}$ of $X_{\mathcal{J}}$ by using Algorithm 1 again. Also note that, if all λ_j, $j \in \mathcal{J}$, are multiple eigenvalues, both of $X'_{\mathcal{J}}$ and $X''_{\mathcal{J}}$ are accurate approximation of the eigenvectors corresponding to the multiple eigenvalues. As a whole, we will derive a refinement algorithm (Algorithm 2 in Section 5), which can deal with clustered eigenvalues.

1.4 Outline

The rest of the paper is organized as follows. In the following section, we summarize the difficulty of the refinement of the eigenvectors corresponding to the clustered eigenvalues, mentioning the previous work relevant to this study, and identify where the higher-precision arithmetic is indispensable. In Section 3, we present a basic refinement algorithm for symmetric eigenvalue decomposition. In Section 4, we provide a convergence analysis of the basic algorithm. On the basis of the basic algorithm, we propose in Section 5 the complete version of a refinement algorithm which can also be applied to matrices having clustered eigenvalues. In Section 6, we present some numerical results showing the behavior and the performance of the proposed algorithms. Finally, we conclude the paper in Section 7.

2 Motivation

Our aim is to develop an iterative refinement algorithm adaptively using the higher-precision arithmetic to obtain arbitrarily accurate eigenvectors. To explain the significance of such an algorithm, we show that it is also effective to obtain the eigenvectors with high accuracy in IEEE 754 binary64 (formerly double precision) floating-point format. Then, $\mathbf{u} \approx 10^{-16}$ as the relative rounding error unit in binary64 format.

2.1 A numerical example

In general, iterative refinement algorithms are important if A has clustered eigenvalues, because it is difficult to compute the eigenvectors corresponding to them accurately. For simplicity, we discuss the iterative refinement for the eigenvectors of

$$A = \begin{bmatrix} 1+\varepsilon & 1 & 1+\varepsilon \\ 1 & 1 & -1 \\ 1+\varepsilon & -1 & 1+\varepsilon \end{bmatrix}. \tag{7}$$

For any ε, the exact eigenvalues and the eigenvector matrix are

$$\begin{cases} \lambda_1 = -1 \\ \lambda_2 = 2 \\ \lambda_3 = 2 + 2\varepsilon \end{cases}, \quad X = \begin{bmatrix} 1/\sqrt{3} & 1/\sqrt{6} & 1/\sqrt{2} \\ -1/\sqrt{3} & 2/\sqrt{6} & 0 \\ -1/\sqrt{3} & -1/\sqrt{6} & 1/\sqrt{2} \end{bmatrix},$$

where A has multiple eigenvalues for $\varepsilon = 0, -3/2$. In what follows, we focus on $|\varepsilon| \ll 1$. Then, $\mathcal{J} = \{2, 3\}, \mathcal{K} = \{1\}$, where $\mathcal{J}, \mathcal{K}$ are defined as in Section 1.3. Since the exact eigenvector matrix X is known, this is a fine example to observe the numerical error of the computed eigenvector matrix $\widehat{X}$. Note that this is not a special example. For other examples having clustered eigenvalues, we obtain similar results.

For $\varepsilon = 2^{-20} \approx 10^{-6}$, the MATLAB built-in function `eig` returns $\widehat{X}, \widehat{D}$, where $\|A\widehat{X} - \widehat{X}\widehat{D}\| \approx 9.62 \times 10^{-16}$, and

$$\|\widehat{X}_{\mathcal{K}} - X_{\mathcal{K}}\| \approx 2.07 \times 10^{-16}, \quad \|\widehat{X}_{\mathcal{J}} - X_{\mathcal{J}}\| \approx 2.69 \times 10^{-11}.$$

We see $\widehat{X}_{\mathcal{J}}$ is not sufficiently close to the exact eigenvector matrix $X_{\mathcal{J}}$ in binary64 format, even though the residual norm $\|A\widehat{X} - \widehat{X}\widehat{D}\|$ is sufficiently small in this example. This result is consistent with the Davis–Kahan theorems as in (3) and (4). In fact, for smaller $\varepsilon = 2^{-50} \approx 10^{-15}$, the function `eig` returns $\widehat{X}$ satisfying

$$\|\widehat{X}_{\mathcal{K}} - X_{\mathcal{K}}\| \approx 1.34 \times 10^{-16}, \quad \|\widehat{X}_{\mathcal{J}} - X_{\mathcal{J}}\| \approx 4.69 \times 10^{-2}.$$

The quality of $\widehat{X}_{\mathcal{J}}$ is seriously poor. However, the residual norm in the right-hand side of (4) is always sufficiently small, whenever eigenvalue problems are solved by backward stable algorithms. Hence, to apply the Jacobi algorithm to $S_{\mathcal{J}}$ in (4) in higher-precision arithmetic is effective. The Davies–Modi algorithm [6] is also promising.

2.2 Previous work

We briefly explain the Davies–Modi algorithm [6], which is relevant to this study. The Davies–Modi algorithm assumes that a real symmetric matrix A is transformed into a nearly diagonal symmetric matrix $S = \widehat{X}^{\mathrm{T}} A \widehat{X}$ by some nearly orthogonal matrix $\widehat{X}$. Under that assumption, the method aims to construct an accurate eigenvector matrix U of S. The idea is seen in Jahn's method [17, 4] as shown by Davies and Smith [7], which derives an SVD algorithm based on the Davies–Modi algorithm. In order to show the relationship between the Davies–Modi and the proposed algorithms, we explain the Davies–Modi algorithm in the same manner as in [7]. We note that U^{T} is written as

$$U^{\mathrm{T}} = e^{Y} = I + Y + \frac{1}{2}Y^2 + \frac{1}{6}Y^3 + \cdots \tag{8}$$

for some skew-symmetric matrix Y. To compute Y with high accuracy, we define a diagonal matrix D_S whose diagonal elements are the eigenvalues of S. Then,

$$D_S = U^{\mathrm{T}} S U = S + YS - SY + \frac{1}{2}(Y^2 S + SY^2) - YSY + \mathcal{O}(\|Y\|^3). \tag{9}$$

Here we define $S_0 = \mathrm{diag}(s_{11}, \ldots, s_{nn})$, and $S_1 = S - S_0$ that corresponds to the off-diagonal entries of S. Under a mild assumption $S_1 = \mathcal{O}(\|Y\|)$, we see

$$\begin{aligned} D_S &= S_0 + (S_1 + YS_0 - S_0 Y) \qquad (10) \\ &\quad + \left(YS_1 - S_1 Y + \frac{1}{2}(Y^2 S_0 + S_0 Y^2) - YS_0 Y \right) + \mathcal{O}(\|Y\|^3). \end{aligned}$$

For the first order approximation, we would like to solve

$$S_1 + \widetilde{Y}_1 S_0 - S_0 \widetilde{Y}_1 = O. \tag{11}$$

If we assume $\widetilde{Y}_1$ is skew-symmetric, $\widetilde{Y}_1$ is easily obtained, namely, $(\widetilde{Y}_1)_{ij} = -(\widetilde{Y}_1)_{ji} = s_{ij}/(s_{ii} - s_{jj})$ for $i \neq j$. To compute U with high accuracy, we construct the second order approximation $\widetilde{U}^{\mathrm{T}} = I + \widetilde{Y}_1 + \widetilde{Y}_2 + \widetilde{Y}_1^2/2$ for skew-symmetric matrices $\widetilde{Y}_1$ and $\widetilde{Y}_2$. To compute $\widetilde{Y}_2$, letting

$$T = \frac{1}{2}(\widetilde{Y}_1 S_1 - S_1 \widetilde{Y}_1), \quad T_0 = \mathrm{diag}(t_{11}, \ldots, t_{nn}), \quad T_1 = T - T_0, \tag{12}$$

we solve $T_1 + \widetilde{Y}_2 S_0 - S_0 \widetilde{Y}_2 = O$ for a skew-symmetric matrix $\widetilde{Y}_2$. Note that $\widetilde{Y}_2$ is computed in the same manner as $\widetilde{Y}_1$ in (11). Thus, we obtain $\widetilde{U}^{\mathrm{T}}$, where its second order approximation is rigorously proved. Since the Davies–Modi algorithm is based on matrix multiplication, it requires $\mathcal{O}(n^3)$ operations.

From now on, we discuss the refinement of $\widehat{X}$ for A in (7), in particular $\widehat{X}_{\mathcal{J}}$ with $\mathcal{J} = \{2, 3\}$, where $\varepsilon = 2^{-20} \approx 10^{-6}$. The main problem is that, since both of the Jacobi and Davies–Modi algorithms are applied to $S = \widehat{X}^{\mathrm{T}} A \widehat{X}$ for computing the eigenvalue decomposition of S, they assume that $\widehat{X}$ is an orthogonal matrix. In general, however, it is not the case, since $\widehat{X}$ suffers from rounding errors, and $\|\widehat{X}^{\mathrm{T}}\widehat{X} - I\| \approx 7.12 \times 10^{-16}$. Therefore, even if the computation of $\widehat{X}^{\mathrm{T}} A \widehat{X}$ is performed in the exact arithmetic, A and S are not similar unless $\widehat{X}$ is orthogonal. Then, it may cause a significant change of the eigenvectors associated with clustered eigenvalues. In fact, even if $X' = \widehat{X}U$ is obtained with U being the eigenvector matrix of S, we have

$$\|X'_{\mathcal{K}} - X_{\mathcal{K}}\| \approx 2.24 \times 10^{-16}, \quad \|X'_{\mathcal{J}} - X_{\mathcal{J}}\| \approx 1.71 \times 10^{-10}.$$

Unfortunately, $X'_{\mathcal{J}}$ is not refined at all as long as the original $\widehat{X}$ involves rounding error in binary64 floating-point arithmetic. To overcome such a problem, we must refine the orthogonality of $\widehat{X}$ as preconditioning in higher-precision arithmetic. Recall that the use of higher-precision arithmetic should be basically restricted to matrix multiplication for much better computational efficiency. Hence, one may use the Newton–Schulz iteration [16, Section 8.3] such as

$$Z = \frac{1}{2}\widehat{X}(3I - \widehat{X}^{\mathrm{T}}\widehat{X}), \tag{13}$$

where all the singular values of $\widehat{X}$ are quadratically convergent to 1. We apply (13) to $\widehat{X}$ in binary128 arithmetic (about 34 digits), and then $\|Z^{\mathrm{T}}Z - I\| \approx 3.81 \times 10^{-31}$. After this reorthogonalization, for the eigenvector matrix U of $T = Z^{\mathrm{T}} A Z$, the columns of $X' := ZU$ are expected to be sufficiently accurate approximation of the eigenvectors of A. In fact, for this theoretical X' rounded into binary64 format, we have

$$\|X'_{\mathcal{K}} - X_{\mathcal{K}}\| \approx 5.79 \times 10^{-17}, \quad \|X'_{\mathcal{J}} - X_{\mathcal{J}}\| \approx 6.84 \times 10^{-17},$$

which means $X' = [X'_{\mathcal{K}}, X'_{\mathcal{J}}]$ is maximally accurate in binary64 format. Of course, in practice, we need to derive a certain method to compute an approximate eigenvector matrix $\widetilde{U}$ of T. If T is nearly diagonal, it is effective to apply a diagonal shift to T. Similarly, the Jacobi and Davies–Modi algorithms would be able to compute $\widetilde{U}$ accurately by the use of higher-precision arithmetic. As a result, we can obtain a sufficiently accurate eigenvector matrix $X' := Z\widetilde{U}$.

In summary, there are two phases for the refinement of eigenvectors. We require the orthogonalization algorithm such as the Newton–Schulz iteration, and the eigenvalue decomposition algorithm such as the Jacobi and the Davies–Modi algorithms, where higher-precision arithmetic is indispensable.

2.3 Position and strength of our algorithm

Clearly, the second equation in (2) corresponds to (9) in the Davies–Modi algorithm. As stated before, the first equation in (2) is required to refine the orthogonality, which corresponds to the Newton–Schulz iteration, and hence our approach is straightforward and convincing. Note that the first order approximation (11) is sufficient in the asymptotic regime because the Newton–Schulz iteration is the first order approximation, which derives quadratic convergence. From this viewpoint, compared to the naive combination of them, we can reduce the number of matrix multiplications in the proposed algorithm. In other words, we propose a new effective algorithm, which can also be interpreted as a sophisticated combination of the Newton–Schulz iteration and the Davies–Modi algorithm. See Remark 2 for details.

In fact, for the example above, the result X' obtained by the proposed algorithm (Algorithm 1) in binary128 arithmetic and rounded into binary64 format satisfies

$$\|X'_{\mathcal{K}} - X_{\mathcal{K}}\| \approx 5.79 \times 10^{-17}, \quad \|X'_{\mathcal{J}} - X_{\mathcal{J}}\| \approx 6.84 \times 10^{-17}.$$

We see $X' = [X'_{\mathcal{K}}, X'_{\mathcal{J}}]$ is sufficiently close to the corresponding exact eigenvector matrix $X = [X_{\mathcal{K}}, X_{\mathcal{J}}]$ in binary64 format. Moreover, for $\varepsilon = 2^{-50} \approx 10^{-15}$, the complete version (Algorithm 2) can iteratively provide accurate approximation of the eigenvectors, up to the limit of computational precision in use. In what follows, we derive the basic algorithm and the complete version in turn.

3 Basic algorithm

Let $A = A^{\mathrm{T}} \in \mathbb{R}^{n\times n}$. The eigenvalues of A are denoted by $\lambda_i \in \mathbb{R}$, $i = 1, \ldots, n$. Then $\|A\| = \max_i |\lambda_i|$. Let $X, \widehat{X} \in \mathbb{R}^{n\times n}$ denote the eigenvector matrix comprising normalized eigenvectors of A and its approximation, respectively. Define $E \in \mathbb{R}^{n\times n}$ such that $X = \widehat{X}(I + E)$. The problem is how to derive the method to compute E. Here, we assume that

$$\|E\| =: \epsilon < 1, \tag{14}$$

which implies that $I + E$ is nonsingular.

First, we consider the orthogonality of the eigenvectors such that $X^{\mathrm{T}}X = I$. From this, we obtain $I = X^{\mathrm{T}}X = (I + E)^{\mathrm{T}}\widehat{X}^{\mathrm{T}}\widehat{X}(I + E)$, and

$$(I + E)^{-\mathrm{T}}(I + E)^{-1} = \widehat{X}^{\mathrm{T}}\widehat{X}. \tag{15}$$

Using the Neumann series expansion, we have

$$(I + E)^{-1} = I - E + \Delta_E, \quad \Delta_E := \sum_{k=2}^{\infty}(-E)^k. \tag{16}$$

Here, it follows from (14) that

$$\|\Delta_E\| \le \frac{\epsilon^2}{1-\epsilon}. \tag{17}$$

Substituting (16) into (15) yields $(I - E + \Delta_E)^{\mathrm{T}}(I - E + \Delta_E) = \widehat{X}^{\mathrm{T}}\widehat{X}$, and

$$E + E^{\mathrm{T}} = I - \widehat{X}^{\mathrm{T}}\widehat{X} + \Delta_1, \tag{18}$$

where $\Delta_1 := \Delta_E + \Delta_E^{\mathrm{T}} + (E - \Delta_E)^{\mathrm{T}}(E - \Delta_E)$. Here it follows from (14) and (17) that

$$\|\Delta_1\| \le \frac{(3-2\epsilon)\epsilon^2}{(1-\epsilon)^2}. \tag{19}$$

Omitting Δ_1 from (18) yields the following matrix equation for $\widetilde{E} = (\widetilde{e}_{ij}) \in \mathbb{R}^{n\times n}$:

$$\widetilde{E} + \widetilde{E}^{\mathrm{T}} = I - \widehat{X}^{\mathrm{T}}\widehat{X}. \tag{20}$$

Next, we consider the diagonalization of A such that $X^{\mathrm{T}}AX = D$. From this, we have $D = X^{\mathrm{T}}AX = (I+E)^{\mathrm{T}}\widehat{X}^{\mathrm{T}}A\widehat{X}(I+E)$, and

$$(I+E)^{-\mathrm{T}}D(I+E)^{-1} = \widehat{X}^{\mathrm{T}}A\widehat{X}. \tag{21}$$

Substitution of (16) into (21) yields $(I - E + \Delta_E)^{\mathrm{T}}D(I - E + \Delta_E) = \widehat{X}^{\mathrm{T}}A\widehat{X}$, and

$$D - DE - E^{\mathrm{T}}D = \widehat{X}^{\mathrm{T}}A\widehat{X} + \Delta_2, \tag{22}$$

where $\Delta_2 := -D\Delta_E - \Delta_E^{\mathrm{T}}D - (E - \Delta_E)^{\mathrm{T}}D(E - \Delta_E)$. Here, it follows from (14) and (17) that

$$\|\Delta_2\| \le \frac{(3-2\epsilon)\epsilon^2}{(1-\epsilon)^2}\|D\| = \frac{(3-2\epsilon)\epsilon^2}{(1-\epsilon)^2}\|A\|. \tag{23}$$

Omission of Δ_2 from (22) yields the following matrix equation for $\widetilde{E} = (\widetilde{e}_{ij}) \in \mathbb{R}^{n\times n}$ and $\widetilde{D} = (\widetilde{d}_{ij}) = \mathrm{diag}(\widetilde{\lambda}_i) \in \mathbb{R}^{n\times n}$:

$$\widetilde{D} - \widetilde{D}\widetilde{E} - \widetilde{E}^{\mathrm{T}}\widetilde{D} = \widehat{X}^{\mathrm{T}}A\widehat{X}. \tag{24}$$

Note that $\widetilde{D}$ is restricted to being diagonal, which implies that $\widetilde{d}_{ii} = \widetilde{\lambda}_i$ and $\widetilde{d}_{ij} = 0$ for $i \ne j$.

Summarizing the above-mentioned discussion with (20) and (24), we obtain the system of matrix equations

$$\left\{\begin{array}{l} \widetilde{E} + \widetilde{E}^{\mathrm{T}} = R \\ \widetilde{D} - \widetilde{D}\widetilde{E} - \widetilde{E}^{\mathrm{T}}\widetilde{D} = S \end{array}\right. \tag{25}$$

$$\Leftrightarrow \quad \left\{\begin{array}{l} \widetilde{e}_{ij} + \widetilde{e}_{ji} = r_{ij} \\ \widetilde{d}_{ij} - \widetilde{\lambda}_i\widetilde{e}_{ij} - \widetilde{\lambda}_j\widetilde{e}_{ji} = s_{ij} \end{array}\right. \qquad (1 \le i, j \le n), \tag{26}$$

where $R = (r_{ij})$ and $S = (s_{ij})$ are defined as $R := I - \widehat{X}^{\mathrm{T}}\widehat{X}$ and $S := \widehat{X}^{\mathrm{T}}A\widehat{X}$, respectively.

Actually, (25) is surprisingly easy to solve. First, we focus on the diagonal parts of $\widetilde{E}$. From the first equation in (26), it follows that $\widetilde{e}_{ii} = r_{ii}/2$ for $1 \leq i \leq n$. Moreover, the second equation in (26) also yields $(1 - 2\widetilde{e}_{ii})\widetilde{\lambda}_i = (1 - r_{ii})\widetilde{\lambda}_i = s_{ii}$. Here, $r_{ii} \ll 1$ due to the assumption that the columns of $\widehat{X}$ are approximately normalized. Thus, we have

$$\widetilde{\lambda}_i = \frac{s_{ii}}{1 - r_{ii}} \quad (1 \leq i \leq n). \tag{27}$$

Note that this is equivalent to the Rayleigh quotient $\widetilde{\lambda}_i = (\widehat{x}_{(i)}^{\mathrm{T}} A \widehat{x}_{(i)})/(\widehat{x}_{(i)}^{\mathrm{T}} \widehat{x}_{(i)})$, where $\widehat{x}_{(i)}$ is the ith column of $\widehat{X}$.

Next, we focus on the off-diagonal parts of $\widetilde{E}$. The combination of (26) and (27) yields

$$\left\{ \begin{array}{l} \widetilde{e}_{ij} + \widetilde{e}_{ji} = r_{ij} \\ \widetilde{\lambda}_i \widetilde{e}_{ij} + \widetilde{\lambda}_j \widetilde{e}_{ji} = -s_{ij} \end{array} \right. \qquad (1 \leq i, j \leq n, i \neq j),$$

which are simply 2 × 2 linear systems. Therefore, if $\widetilde{\lambda}_i \neq \widetilde{\lambda}_j$, then we have

$$\widetilde{e}_{ij} = \frac{s_{ij} + \widetilde{\lambda}_j r_{ij}}{\widetilde{\lambda}_j - \widetilde{\lambda}_i}.$$

Otherwise, from the first equation in (26) we choose $\widetilde{e}_{ij}$ as

$$\widetilde{e}_{ij} = \widetilde{e}_{ji} = \frac{r_{ij}}{2}.$$

Similar to the Newton–Schulz iteration as mentioned in Section 2, this choice is quite reasonable for improving the orthogonality of $\widehat{X}$ corresponding to $\widehat{x}_{(i)}$ and $\widehat{x}_{(j)}$, which are the ith and jth columns of $\widehat{X}$. Note that, in general, this does not improve the accuracy of $\widehat{x}_{(i)}$ or $\widehat{x}_{(j)}$, unless λ_i and λ_j are exactly multiple eigenvalues.

It is not possible to determine whether A has multiple eigenvalues by using numerical computations, since computed results suffer from rounding errors as a result of finite precision arithmetic. Instead, we regard λ_i and λ_j as clustered eigenvalues if $|\widetilde{\lambda}_i - \widetilde{\lambda}_j| \leq \delta$ for adequate δ. In our algorithm we set $\delta := \rho \cdot \max_{i \neq j} |s_{ij}|$ with a parameter $\rho \geq 1$, where $S = \widehat{X}^{\mathrm{T}}A\widehat{X}$. The parameter ρ relaxes the criterion for judging whether λ_i and λ_j are clustered eigenvalues. The larger ρ is, the more safely the algorithm works. Namely, the use of ρ may avoid the presence of large magnitude values in $\widetilde{E}$, and then the orthogonality of $\widehat{X}$ is improved. A side effect of introducing ρ will appear in our convergence theorem (Theorem 1), which is reflected in the sufficient condition (55) for quadratic convergence. The justification for the choice of δ will be explained in the end of this section.

In Algorithm 1, we present a refinement algorithm for the eigenvalue decomposition of a real symmetric matrix, which is designed to be iteratively applied.

Algorithm 1 RefSyEv: Refinement of approximate eigenvectors of a real symmetric matrix.

```
Require: $A = A^{\mathrm{T}} \in \mathbb{R}^{n\times n}$;  $\widehat{X} \in \mathbb{R}^{n\times \ell}$;  $\rho \in \mathbb{R}$ with $\rho \geq 1$
Ensure: $X' \in \mathbb{R}^{n\times \ell}$;  $e_{\max} \in \mathbb{R}$;  $\widetilde{\lambda} = (\widetilde{\lambda}_1, \ldots, \widetilde{\lambda}_\ell)$;  $\delta \in \mathbb{R}$
1: function $[X', e_{\max}, \widetilde{\lambda}, \delta] \leftarrow$ RefSyEv($A$, $\widehat{X}$, $\rho$)
2:     $R \leftarrow I - \widehat{X}^{\mathrm{T}}\widehat{X}$
3:     $S \leftarrow \widehat{X}^{\mathrm{T}}A\widehat{X}$
4:     $\widetilde{\lambda}_i \leftarrow s_{ii}/(1 - r_{ii})$    for $i = 1, \ldots, \ell$        ▷ Compute approximate eigenvalues.
5:     $\delta \leftarrow \rho \cdot \max_{i\neq j} |s_{ij}|$
6:     $\widetilde{e}_{ij} \leftarrow \begin{cases} r_{ij}/2 & (|\widetilde{\lambda}_i - \widetilde{\lambda}_j| \leq \delta) \\ \dfrac{s_{ij} + \widetilde{\lambda}_j r_{ij}}{\widetilde{\lambda}_j - \widetilde{\lambda}_i} & \text{(otherwise)} \end{cases}$   for $1 \leq i, j \leq \ell$▷ Compute $\widetilde{E}$.
7:     $e_{\max} \leftarrow \max_{i,j} |\widetilde{e}_{ij}|$
8:     $X' \leftarrow \widehat{X} + \widehat{X}\widetilde{E}$        ▷ Update $\widehat{X}$ by $\widehat{X}(I + \widetilde{E})$.
9: end function
```

To show the behavior of Algorithm 1, we will present Theorem 1 in Section 4 that states the quadratic convergence of the algorithm, if all the eigenvalues are simple and well separated. In the case where A has multiple eigenvalues, we obtain a result similar to Theorem 1 by our convergence analysis as Theorem 2. For details, see Section 4. To see the numerical behavior of Algorithm 1, we will present several examples in Section 6, which demonstrate excellent performance of the algorithm.

Let us consider the most likely scenario where $\widehat{X}$ is computed by some backward stable algorithm in ordinary floating-point arithmetic with the relative rounding error unit $\mathbf{u}$. Suppose all the eigenvalues are well separated. Define E and E' such that $X = \widehat{X}(I + E) = X'(I + E')$, where X' is obtained by Algorithm 1. Then, the Davis–Kahan theorem for eigenpairs [22, Theorem 11.7.1] suggests that $\|E\| = \mathcal{O}(\beta\mathbf{u})$, where β is the reciprocal of the maximum relative gap of eigenvalues defined as in (6). As will be stated in Remark 4 in Section 4, $\|E'\| = \mathcal{O}(\max(\beta\mathbf{u}_h, \beta^3\mathbf{u}^2))$. Thus, we should set $\mathbf{u}_h$ to not greater than $\beta^2\mathbf{u}^2$ in order to achieve quadratic convergence. This will also be confirmed numerically in Section 6.

Remark 1. *For the generalized symmetric definite eigenvalue problem $Ax = \lambda Bx$ where A and B are real symmetric with B being positive definite, a similar algorithm can readily be derived by replacing the line 2 in Algorithm 1 by $R \leftarrow I - \widehat{X}^{\mathrm{T}}B\widehat{X}$.*

Remark 2. *It is worth noting that our approach can reproduce the first order approximation step in the Davies–Modi algorithm as follows. Recall that D_S is a diagonal matrix whose diagonal entries are the eigenvalues of S and U is the eigenvector matrix of S. Since we see*

$$\widehat{X}^{\mathrm{T}} A \widehat{X} = S = U D_S U^{\mathrm{T}} = e^{-Y} D_S e^{Y} = D_S + D_S Y - Y D_S + \mathcal{O}(\|Y\|^2) \quad (28)$$

from (8)*, we obtain $S = \widetilde{D}_S + \widetilde{D}_S \widetilde{Y}_1 - \widetilde{Y}_1 \widetilde{D}_S$ as the linearization process in the same manner as* (24)*. It is easy to see that $\widetilde{D}_S = \mathrm{diag}(s_{11}, \ldots, s_{nn})$ and $\widetilde{Y}_1$ is equal to the solution in* (11)*. Hence, if $\widehat{X}$ in Algorithm 1 is an orthogonal matrix and the diagonal elements of S are sufficiently separated, then $\widetilde{E}$ is equal to the skew-symmetric matrix $\widetilde{Y}_1$ in view of $R = O$. In other words, from the viewpoint of the Davies–Modi algorithm, the second order approximation step is removed and the skew-symmetry of $\widetilde{E}$ is not assumed in Algorithm 1. Instead, the condition* (20) *is integrated to improve the orthogonality. If we assume $\widetilde{E} = \widetilde{E}^{\mathrm{T}}$ in* (20)*, we have*

$$X' = \widehat{X}(I + \widetilde{E}) = \widehat{X}\left(I + \frac{1}{2}(I - \widehat{X}^{\mathrm{T}}\widehat{X})\right), \quad (29)$$

which is equivalent to the Newton–Schulz iteration (13)*. In other words, if all the eigenvalues are regarded to be clustered, $\widehat{X}$ is refined by the Newton–Schulz iteration. For the orthogonality, we require the relation* (20) *only, while an iteration function for the polar decomposition must be an odd function such as the Newton–Schulz iteration. In summary, to combine the Newton–Schulz iteration and the Davies–Modi algorithm sophisticatedly, we remove unnecessary conditions from both of them, resulting in Algorithm 1 viewed as an ideal method to refine the orthogonality and the diagonality simultaneously.*

Remark 3. *We mention what happens if the algorithm is applied to not all but a few approximate eigenvectors of A corresponding to well-separated eigenvalues as an input. In this case, the algorithm does not work as refinement of eigenvectors in terms of accuracy. For any $\ell \in \mathbb{N}$, $2 \le \ell < n$, let $\widehat{X}_\ell \in \mathbb{R}^{n \times \ell}$ comprise any ℓ columns of $\widehat{X}$. When we iteratively apply Algorithm 1 to A and $\widehat{X}_\ell$, the orthogonality of the columns of $\widehat{X}_\ell$ is improved by the iterations, i.e., $\widehat{X}_\ell^{\mathrm{T}} \widehat{X}_\ell \to I$, and $\widehat{X}_\ell^{\mathrm{T}} A \widehat{X}_\ell$ converges to some diagonal matrix. However, this does not necessarily imply that the columns of $\widehat{X}_\ell$ converge to the eigenvectors of A. For that result, we must use all approximate eigenvectors of A as an input of the algorithm when dealing with well-separated eigenvalues. For clustered eigenvalues, we will consider the case $\ell < n$ to improve an approximation of the eigenspace corresponding to the clustered eigenvalues in Section 5.*

4 Convergence analysis

In this section, we prove quadratic convergence of Algorithm 1, that is the basic part of the proposed algorithm, on the assumption that the approximate solutions are modestly close to the exact solutions. Our analysis is divided into two parts. First, if we assume that A does not have multiple eigenvalues, then quadratic convergence is proved. Next, we move on to general analysis to any A. More specifically, if the multiple eigenvalues can be identified using δ in Algorithm 1, the quadratic convergence also follows.

Recall that the error of the approximate solution is expressed as $\|\widehat{X} - X\| = \|\widehat{X}E\|$ in view of $X = \widehat{X}(I+E)$. The refined approximate solution is $X' := \widehat{X}(I+\widetilde{E})$. It then follows that the error of the refined solution is expressed as follows:

$$\|\widehat{X}(I+\widetilde{E}) - X\| = \|\widehat{X}(\widetilde{E} - E)\|.$$

In addition, recall that $\widetilde{E}$ is the solution of the following equations:

$$\widetilde{E} + \widetilde{E}^{\mathrm{T}} = R, \tag{30}$$

$$\widetilde{D} - \widetilde{D}\widetilde{E} - \widetilde{E}^{\mathrm{T}}\widetilde{D} = S, \tag{31}$$

where

$$R := I - \widehat{X}^{\mathrm{T}}\widehat{X}, \tag{32}$$

$$S := \widehat{X}^{\mathrm{T}}A\widehat{X}. \tag{33}$$

However, if $\widetilde{\lambda}_i \approx \widetilde{\lambda}_j$, then (31) is not reflected for the computation of $\widetilde{e}_{ij}$ and $\widetilde{e}_{ji}$. In this case, we choose $\widetilde{e}_{ij} = \widetilde{e}_{ji} = r_{ij}/2$ from (30). Such an exceptional case is considered later in the subsection on multiple eigenvalues.

Briefly, our goal is to prove quadratic convergence

$$\|\widehat{X}(I+\widetilde{E}) - X\| = \mathcal{O}(\|\widehat{X} - X\|^2),$$

which corresponds to

$$\|\widehat{X}(\widetilde{E} - E)\| = \mathcal{O}(\|\widehat{X}E\|^2),$$

as $\widehat{X} \to X$. We would like to prove that

$$\|\widetilde{E} - E\| = \mathcal{O}(\|E\|^2) \tag{34}$$

as $\|E\| \to 0$.

To investigate the relationship between E and $\widetilde{E}$, let

$$\epsilon := \|E\|,$$

$$\chi(\epsilon) := \frac{3-2\epsilon}{(1-\epsilon)^2}. \tag{35}$$

Then, we see that

$$E + E^{\mathrm{T}} = R + \Delta_1, \quad \|\Delta_1\| \leq \chi(\epsilon)\epsilon^2 \tag{36}$$

from (18) and (19). In addition, we have

$$D - DE - E^{\mathrm{T}}D = S + \Delta_2, \quad \|\Delta_2\| \leq \chi(\epsilon)\|A\|\epsilon^2 \tag{37}$$

from (22) and (23).

4.1 Simple eigenvalues

First, we focus on the situation where the eigenvalues are simple and well separated. Here, we derive a key lemma that shows (34).

Lemma 1. *Let A be a real symmetric $n \times n$ matrix. Assume that all the eigenvalues of A are simple. Suppose that Algorithm 1 is applied to A and $\widehat{X} \in \mathbb{R}^{n\times n}$ with $\rho \geq 1$, where δ is determined by*

$$S := \widehat{X}^{\mathrm{T}} A \widehat{X}, \quad \delta := \rho \cdot \max_{i\neq j} |s_{ij}|. \tag{38}$$

Moreover, suppose that

$$\epsilon < \frac{1}{\rho} \min\left(\frac{\min_{i\neq j} |\lambda_i - \lambda_j|}{10n\|A\|}, \frac{1}{100} \right), \tag{39}$$

where $\epsilon := \|E\|$ and $X = \widehat{X}(I + E)$. Then, we obtain

$$|\widetilde{e}_{ii} - e_{ii}| \leq \frac{\chi(\epsilon)}{2}\epsilon^2 \quad (i = 1, \dots, n) \tag{40}$$

$$|\widetilde{e}_{ij} - e_{ij}| \leq \frac{(2\chi(\epsilon) + 21\epsilon)\|A\|\epsilon^2}{|\lambda_i - \lambda_j| - 14\|A\|\epsilon^2} \quad (i \neq j) \tag{41}$$

Proof. First, we estimate the diagonal elements. It is easy to see that

$$(E - \widetilde{E}) + (E - \widetilde{E})^{\mathrm{T}} = \Delta_1, \quad \|\Delta_1\| \leq \chi(\epsilon)\epsilon^2 \tag{42}$$

from (30), (32), and (36). Hence, we obtain (40) from the diagonal elements in (42).

Next, we discuss $\widetilde{D}$. From (31) and (37), $\widetilde{D}$ and D are determined as $\widetilde{\lambda}_i = s_{ii}/(1 - 2\widetilde{e}_{ii})$, $\lambda_i = (s_{ii} + \Delta_2(i,i))/(1 - 2e_{ii})$. Thus, we have

$$\begin{aligned} \widetilde{\lambda}_i - \lambda_i &= \frac{s_{ii}(1 - 2e_{ii}) - (s_{ii} + \Delta_2(i,i))(1 - 2\widetilde{e}_{ii})}{(1 - 2e_{ii})(1 - 2\widetilde{e}_{ii})} \\ &= -\frac{(1 - 2\widetilde{e}_{ii})\Delta_2(i,i) + 2(e_{ii} - \widetilde{e}_{ii})s_{ii}}{(1 - 2e_{ii})(1 - 2\widetilde{e}_{ii})} \\ &= -\frac{\Delta_2(i,i)}{1 - 2e_{ii}} - \frac{2(e_{ii} - \widetilde{e}_{ii})s_{ii}}{(1 - 2e_{ii})(1 - 2\widetilde{e}_{ii})}. \end{aligned} \tag{43}$$

For the first term on the right-hand side, we see

$$\left|\frac{\Delta_2(i,i)}{1-2e_{ii}}\right| < \frac{\chi(\epsilon)}{1-2e_{ii}}\|A\|\epsilon^2$$

from (37). Moreover, for the second term,

$$\left|\frac{2(e_{ii}-\widetilde{e}_{ii})s_{ii}}{(1-2e_{ii})(1-2\widetilde{e}_{ii})}\right| < \frac{(1+2\epsilon+\chi(\epsilon)\epsilon^2)\chi(\epsilon)}{(1-2e_{ii})(1-2\widetilde{e}_{ii})}\|A\|\epsilon^2$$

from (33), (37), and (40). In addition, from (35) and $\epsilon < 1/100$ in (55), we have

$$\chi(\epsilon) = \frac{3-2\epsilon}{(1-\epsilon)^2} \leq 3.0405\cdots . \tag{44}$$

Noting (43) and

$$\begin{aligned} & \frac{\chi(\epsilon)}{1-2e_{ii}}\|A\|\epsilon^2 + \frac{(1+2\epsilon+\chi(\epsilon)\epsilon^2)\chi(\epsilon)}{(1-2e_{ii})(1-2\widetilde{e}_{ii})}\|A\|\epsilon^2 \\ = \;& \frac{(1-2\widetilde{e}_{ii})+(1+2\epsilon+\chi(\epsilon)\epsilon^2)}{(1-2e_{ii})(1-2\widetilde{e}_{ii})}\chi(\epsilon)\|A\|\epsilon^2 \\ \leq \;& \frac{2(1+2\epsilon+\chi(\epsilon)\epsilon^2)\chi(\epsilon)}{(1-2\epsilon)(1-2\epsilon-\chi(\epsilon)\epsilon^2)}\|A\|\epsilon^2 \\ < \;& 7\|A\|\epsilon^2, \end{aligned}$$

we have

$$|\widetilde{\lambda}_i - \lambda_i| < 7\|A\|\epsilon^2 \quad (i = 1,\ldots,n). \tag{45}$$

In what follows, we estimate the off-diagonal elements of $\widetilde{E}$. Combining (37) with (45), we have

$$\widetilde{D} - \widetilde{D}E - E^{\mathrm{T}}\widetilde{D} = S + \widetilde{\Delta}_2,$$

where off-diagonal elements of $|\Delta_2 - \widetilde{\Delta}_2|$ are less than $14\|A\|\epsilon^3$. In other words,

$$|\widetilde{\Delta}_2(i,j)| \quad \leq \quad (\chi(\epsilon)+14\epsilon)\|A\|\epsilon^2 \tag{46}$$

for $i \neq j$ from (37), where $\widetilde{\Delta}_2(i,j)$ are the (i,j) elements of $\widetilde{\Delta}_2$. In addition, from (31), it follows that

$$\widetilde{D}(E-\widetilde{E}) + (E-\widetilde{E})^{\mathrm{T}}\widetilde{D} = -\widetilde{\Delta}_2. \tag{47}$$

Combining (47) with (42), we estimate the off-diagonal elements of $\widetilde{E}$.

For all $i \neq j$, we derive

$$\begin{aligned}\rho|s_{ij}| &\leq \rho((|\lambda_i| + |\lambda_j|)\epsilon + \chi(\epsilon)\|A\|\epsilon^2) \\ &\leq \rho(2 + \chi(\epsilon)\epsilon)\|A\|\epsilon \\ &< \frac{\min_{i\neq j}|\lambda_i - \lambda_j|}{2} \\ &< \min_{i\neq j}|\lambda_i - \lambda_j| - 15\|A\|\epsilon^2,\end{aligned}$$

where the first, third, and forth inequalities are due to (37), (55) and (44), and $\|A\|\|E\|^2 < \min_{i\neq j}|\lambda_i - \lambda_j|/1000$ obtained from (55), respectively. Noting (45), we find that

$$|\widetilde{\lambda}_i - \widetilde{\lambda}_j| \geq |\lambda_i - \lambda_j| - 14\|A\|\epsilon^2 > \delta \ (= \rho \max_{i\neq j}|s_{ij}|).$$

Hence, we focus on the linear system corresponding to such i, j. From (42), (46) and (47), we have

$$(e_{ij} - \widetilde{e}_{ij}) + (e_{ji} - \widetilde{e}_{ji}) = \epsilon_1, \quad |\epsilon_1| \leq \chi(\epsilon)\epsilon^2, \tag{48}$$

$$\widetilde{\lambda}_i(e_{ij} - \widetilde{e}_{ij}) + \widetilde{\lambda}_j(e_{ji} - \widetilde{e}_{ji}) = \epsilon_2, \quad |\epsilon_2| \leq (\chi(\epsilon) + 14\epsilon)\|A\|\epsilon^2. \tag{49}$$

It then follows that

$$e_{ij} - \widetilde{e}_{ij} = \frac{\epsilon_2 - \widetilde{\lambda}_j\epsilon_1}{\widetilde{\lambda}_i - \widetilde{\lambda}_j}, \quad e_{ji} - \widetilde{e}_{ji} = \frac{\epsilon_2 - \widetilde{\lambda}_i\epsilon_1}{\widetilde{\lambda}_j - \widetilde{\lambda}_i}.$$

Therefore, using (45), we obtain

$$|\widetilde{e}_{ij} - e_{ij}| \leq \frac{(2\chi(\epsilon) + 21\epsilon)\|A\|\epsilon^2}{|\widetilde{\lambda}_i - \widetilde{\lambda}_j|} \leq \frac{(2\chi(\epsilon) + 21\epsilon)\|A\|\epsilon^2}{|\lambda_i - \lambda_j| - 14\|A\|\epsilon^2} \tag{50}$$

□

From Lemma 1, the following two lemmas are readily accessible.

Lemma 2. *Under the same assumptions as for Lemma 1, we have*

$$\|\widetilde{E} - E\| < \frac{7}{10}\epsilon. \tag{51}$$

Proof. In view of $\|\widetilde{E} - E\|^2 \leq \sum_{i,j}|\widetilde{e}_{ij} - e_{ij}|^2$ in (40) and (41), we obtain

$$\|\widetilde{E} - E\| \leq \frac{(2\chi(\epsilon) + 21\epsilon)n\|A\|\epsilon^2}{\min_{i\neq j}|\lambda_i - \lambda_j| - 14\|A\|\epsilon^2}. \tag{52}$$

From (44), we have

$$\|\widetilde{E} - E\| < \frac{6.4n\|A\|\epsilon^2}{\min_{i\neq j}|\lambda_i - \lambda_j| - 14\|A\|\epsilon^2} < \frac{6.4\epsilon}{10(1 - \frac{2}{100n})} < \frac{7}{10}\epsilon. \tag{53}$$

□

Lemma 3. *Under the same assumptions as for Lemma 1, we have*

$$\limsup_{\epsilon\to 0}\frac{\|\widetilde{E}-E\|}{\epsilon^2}\le\frac{6n\|A\|}{\min_{i\neq j}|\lambda_i-\lambda_j|}. \tag{54}$$

Proof. From (35), we have $\chi(0)=3$. Combining that with (52), we obtain (54). □

Based on the above lemmas, we obtain a main theorem that states the quadratic convergence of Algorithm 1, if all the eigenvalues are simple and well-separated.

Theorem 1. *Let A be a real symmetric $n\times n$ matrix. Assume that all the eigenvalues λ_i of A are simple. Suppose that Algorithm 1 is applied to A and $\widehat{X}\in\mathbb{R}^{n\times n}$ for some $\rho\ge 1$, and X' is obtained. Define E and E' such that $X=\widehat{X}(I+E)$ and $X=X'(I+E')$, respectively. If*

$$\|E\|<\frac{1}{\rho}\min\left(\frac{\min_{i\neq j}|\lambda_i-\lambda_j|}{10n\|A\|},\frac{1}{100}\right), \tag{55}$$

then

$$\|E'\|<\frac{5}{7}\|E\|, \tag{56}$$

$$\limsup_{\|E\|\to 0}\frac{\|E'\|}{\|E\|^2}\le\frac{6n\|A\|}{\min_{i\neq j}|\lambda_i-\lambda_j|}. \tag{57}$$

Proof. Noting $X'(I+E')=\widehat{X}(I+E)$ $(=X)$, we have

$$X'E'=\widehat{X}(E-\widetilde{E}).$$

Therefore, we obtain

$$E'=(I+\widetilde{E})^{-1}(E-\widetilde{E}). \tag{58}$$

Noting $\|\widetilde{E}\|\le 1/50$ from Lemma 1 and (51), we have

$$\|E'\|\le\frac{\|\widetilde{E}-E\|}{1-\|\widetilde{E}\|}<\frac{5}{7}\|E\|. \tag{59}$$

Finally, using (58) and (54), we obtain (57). □

Remark 4. *In practice, the accuracy of a refined eigenvector matrix X' is restricted by the computational precision being used. Since $\widehat{X}$ is improved quadratically in the exact arithmetic, the computational precision used in the algorithm must correspond to $\|E\|^2$ for preserving the convergence property of the algorithm. In general, if Algorithm 1 is performed in higher-precision* *arithmetic with the relative rounding error unit* $\mathbf{u}_h$*, then due to the rounding errors,*

$$\|E'\| = \mathcal{O}(\max(\beta\mathbf{u}_h, \beta\|E\|^2)), \quad \beta := \frac{\|A\|}{\min_{\lambda_i \neq \lambda_j} |\lambda_i - \lambda_j|}. \tag{60}$$

This is obtained as follows. Let $\widehat{R}$*,* $\widehat{S}$*,* $\widehat{E}$*,* $\widehat{X}'$ *denote the computed results of* R*,* S*,* $\widetilde{E}$*,* X'*, respectively. Then, define* E' *such that* $X = \widehat{X}'(I + E')$*. From a standard rounding error analysis, we have*

$$\begin{aligned} \widehat{R} &= R + \Delta_R, \quad \|\Delta_R\| = \mathcal{O}(\mathbf{u}_h), \\ \widehat{S} &= S + \Delta_S, \quad \|\Delta_S\| = \mathcal{O}(\|A\|\mathbf{u}_h), \\ \widehat{E} &= \widetilde{E} + \Delta_E, \end{aligned}$$

where

$$\|\Delta_E\| = \|\widehat{E} - \widetilde{E}\| \leq \frac{\|\Delta_S\| + \|A\|\|\Delta_R\|}{\min_{\lambda_i \neq \lambda_j} |\lambda_i - \lambda_j|}(1 + \mathcal{O}(\beta\|E\|^2)) = \mathcal{O}(\beta\mathbf{u}_h), \tag{61}$$

due to line 6 in Algorithm 1. Then, combining (57) *and* (61) *yields* (60)*.*

4.2 Multiple eigenvalues

Multiple eigenvalues require some care. We might not be able to solve the linear system given by (30) and (31) in the case $\widetilde{\lambda}_i \approx \widetilde{\lambda}_j$ corresponding to multiple eigenvalues $\lambda_i = \lambda_j$, and hence, we use the first equation (30) only, i.e., $\widetilde{e}_{ij} = \widetilde{e}_{ji} = r_{ij}/2$ for $|\widetilde{\lambda}_i - \widetilde{\lambda}_j| \leq \delta$. This exceptional process is relevant to the definitions of X and E. Suppose λ_i, $i \in \mathcal{M} := \{i_1, i_2, \ldots, i_p\}$, are multiple eigenvalues. Then, the i_kth columns of X for $1 \leq k \leq p$ are not unique: for $X_{\mathcal{M}} := [x_{(i_1)}, \ldots, x_{(i_p)}] \in \mathbb{R}^{n \times p}$, the columns of $X_{\mathcal{M}}Q$ are also eigenvectors of A for any orthogonal matrix $Q \in \mathbb{R}^{p \times p}$. Hence, let $\{X\}$ be the set of the $n \times n$ eigenvector matrices of A, and $\{E\} := \{\widehat{X}^{-1}X - I : X \in \{X\}\}$.

The key idea of the proof of quadratic convergence below is to define an eigenvector matrix $Y \in \{X\}$ as follows. For any $X_\alpha \in \{X\}$, there is a permutation matrix P such that $X_\alpha P = [X_{\mathcal{M}}, X_{\mathcal{K}}]$. Suppose an approximation $\widehat{X}$ of X_α is nonsingular. Then, splitting $(\widehat{X}P)^{-1}X_{\mathcal{M}}$ into the first p rows $V_\alpha \in \mathbb{R}^{p \times p}$ and the rest $(n-p)$ rows $W_\alpha \in \mathbb{R}^{(n-p) \times p}$, we have

$$(\widehat{X}P)^{-1}X_{\mathcal{M}} = P^{\mathrm{T}}\widehat{X}^{-1}X_{\mathcal{M}} = \begin{bmatrix} V_\alpha \\ W_\alpha \end{bmatrix} = \begin{bmatrix} C \\ W_\alpha Q_\alpha^{\mathrm{T}} \end{bmatrix} Q_\alpha$$

in view of the polar decomposition $V_\alpha = CQ_\alpha$, where $C \in \mathbb{R}^{p \times p}$ is symmetric and positive definite, and $Q_\alpha \in \mathbb{R}^{p \times p}$ is orthogonal. Note that, although $X_{\mathcal{M}}Q$ for any orthogonal matrix $Q \in \mathbb{R}^{p \times p}$ is also the eigenvector matrix, the symmetric positive definite matrix C is independent of Q. In other words, we have

$$(\widehat{X}P)^{-1}(X_{\mathcal{M}}Q) = P^{\mathrm{T}}\widehat{X}^{-1}(X_{\mathcal{M}}Q) = \left[\begin{array}{c} V_{\alpha}Q \\ W_{\alpha}Q \end{array}\right] = \left[\begin{array}{c} C \\ W_{\alpha}Q_{\alpha}^{\mathrm{T}} \end{array}\right] Q_{\alpha}Q,$$

where $V_{\alpha}Q = C(Q_{\alpha}Q)$ is the unique polar decomposition of $V_{\alpha}Q$. Hence, we define the unique matrix $Y := [X_{\mathcal{M}}Q_{\alpha}^{\mathrm{T}}, X_{\mathcal{K}}]P^{\mathrm{T}}$ for all the matrices in $\{X\}$, where Y depends on $\widehat{X}$ only. Then, the corresponding error term $F = (f_{ij})$ is uniquely determined as

$$\begin{aligned} F &:= \widehat{X}^{-1}Y - I = [\widehat{X}^{-1}X_{\mathcal{M}}Q_{\alpha}^{\mathrm{T}}, \widehat{X}^{-1}X_{\mathcal{K}}]P^{\mathrm{T}} - I \\ &= P\left[\begin{array}{cc} C - I & * \\ * & * \end{array}\right]P^{\mathrm{T}}, \end{aligned}$$

which implies $f_{ij} = f_{ji}$ corresponding to the multiple eigenvalues $\lambda_i = \lambda_j$. Therefore,

$$f_{ij} = f_{ji} = \frac{r_{ij} + \Delta_1(i,j)}{2} \tag{62}$$

from (36), where $\Delta_1(i,j)$ denote (i,j) elements of Δ_1 for all i,j. Also note that, if $\widehat{X}$ is an exact eigenvector matrix, $\|F\| = 0$ holds. Our aim is to prove $\|F\| \to 0$ in the iterative refinement for $\widehat{X} \approx Y \in \{X\}$, where Y depends on $\widehat{X}$. To this end, for the refined X', we also define an eigenvector matrix $Y' \in \{X\}$ and $F' := (X')^{-1}Y' - I$ such that the submatrices of $(X')^{-1}Y'$ corresponding to the multiple eigenvalues are symmetric positive definite. Note that the eigenvector matrix Y is changed to Y' corresponding to X' after the refinement. For the convergence analysis, define the index sets $\mathcal{M}_k$, $k = 1, 2, \ldots, M$, for multiple eigenvalues $\{\lambda_i\}_{i\in\mathcal{M}_k}$ satisfying the following conditions:

$$\left\{\begin{array}{l} \text{(a) } \mathcal{M}_k \subseteq \{1,2,\ldots,n\} \text{ with } n_k := |\mathcal{M}_k| \geq 2 \\ \text{(b) } \lambda_i = \lambda_j,\ \forall i,j \in \mathcal{M}_k \\ \text{(c) } \lambda_i \neq \lambda_j,\ \forall i \in \mathcal{M}_k,\ \forall j \in \{1,2,\ldots,n\} \setminus \mathcal{M}_k \end{array}\right. . \tag{63}$$

Using the above definitions, we obtain the following key lemma to prove quadratic convergence.

Lemma 4. *Let A be a real symmetric $n \times n$ matrix with multiple eigenvalues and the index sets $\mathcal{M}_k$, $k = 1, 2, \ldots, M$ satisfy* (63)*. For a given $\widehat{X} \in \mathbb{R}^{n\times n}$, let $\{X\}$ be the set of the $n \times n$ eigenvector matrices for A and*

$$\{E\} := \{\widehat{X}^{-1}X - I : X \in \{X\}\}. \tag{64}$$

Then, there exists a unique $Y \in \{X\}$ such that, for all k, the $n_k \times n_k$ submatrices of $\widehat{X}^{-1}Y$ corresponding to $\{\lambda_i\}_{i\in\mathcal{M}_k}$ be symmetric and positive *definite. Furthermore, define $F \in \{E\}$ such that $Y = \widehat{X}(I + F)$. Then, for any $E \in \{E\}$,*

$$\|F\| \leq 3\|E\|. \tag{65}$$

Proof. For any E in (64), let E_{diag} denote the block diagonal part of E whose $n_k \times n_k$ block corresponds to n_k multiple eigenvalues, i.e.,

$$E_{\mathrm{diag}}(i,j) := \begin{cases} e_{ij} & (\lambda_i = \lambda_j) \\ 0 & (\text{otherwise}) \end{cases}$$

where $\mathrm{diag}(\lambda_1, \ldots, \lambda_n) = X^{\mathrm{T}}AX$. Here, we consider the polar decomposition

$$I + E_{\mathrm{diag}} =: HU, \tag{66}$$

where H is a symmetric positive definite matrix, and U is an orthogonal matrix. Then, we have

$$Y = XU^{\mathrm{T}} \tag{67}$$

from the definition of Y and

$$\begin{aligned} F &= \widehat{X}^{-1}Y - I \\ &= \widehat{X}^{-1}XU^{\mathrm{T}} - I \\ &= (E + I)U^{\mathrm{T}} - I \\ &= (E - E_{\mathrm{diag}} + HU)U^{\mathrm{T}} - I \\ &= (E - E_{\mathrm{diag}})U^{\mathrm{T}} + H - I, \end{aligned} \tag{68}$$

where the first, second, third, and fourth equalities are consequences of the definition of F, (67), (64), and (66), respectively. In addition, we see that

$$\|H - I\| \leq \|E_{\mathrm{diag}}\| \tag{69}$$

because all the eigenvalues of H range over the interval $[1 - \|E_{\mathrm{diag}}\|, 1 + \|E_{\mathrm{diag}}\|]$ from (66). In addition, note that

$$\|E_{\mathrm{diag}}\| \leq \|E\|. \tag{70}$$

Therefore, we obtain

$$\|F\| = \|(E - E_{\mathrm{diag}})U^{\mathrm{T}} + (H - I)\| \leq 3\|E\| \tag{71}$$

from (68), (69), and (70), giving us (65). □

On the basis of Theorem 1 and Lemma 4, we see the quadratic convergence for a real symmetric matrix A that has multiple eigenvalues.

Theorem 2. *Let A be a real symmetric $n \times n$ matrix with multiple eigenvalues and the index sets $\mathcal{M}_k$, $k = 1, 2, \ldots, M$ satisfy* (63)*. In addition, let $\{X\}$ be the set of the $n \times n$ eigenvector matrices for A. Suppose that Algorithm 1 is applied to A and $\widehat{X} \in \mathbb{R}^{n\times n}$ for some $\rho \geq 1$, and X' is obtained. Define $S = (s_{ij}) \in \mathbb{R}^{n\times n}$ such that $S = \widehat{X}^{\mathrm{T}} A \widehat{X}$, and*

$$\delta := \rho \cdot \max_{i \neq j} |s_{ij}|.$$

For all k, let the $n_k \times n_k$ submatrices of $\widehat{X}^{-1}Y$ and $(X')^{-1}Y'$ corresponding to $\{\lambda_i\}_{i\in\mathcal{M}_k}$ be symmetric and positive definite, where $Y, Y' \in \{X\}$. Furthermore, define F and F' such that $Y = \widehat{X}(I + F)$ and $Y' = X'(I + F')$, respectively. Suppose that

$$\epsilon_F := \|F\| < \frac{1}{3\rho} \min\left(\frac{\min_{\lambda_i\neq\lambda_j} |\lambda_i - \lambda_j|}{10n\|A\|}, \frac{1}{100}\right). \tag{72}$$

In addition, assume that

$$14\|A\|\|F\|^2 < \delta. \tag{73}$$

Then, we obtain

$$\|F'\| < \frac{5}{7}\|F\|, \tag{74}$$

$$\limsup_{\|F\|\to 0} \frac{\|F'\|}{\|F\|^2} \leq 3\left(\frac{6n\|A\|}{\min_{\lambda_i\neq\lambda_j} |\lambda_i - \lambda_j|}\right). \tag{75}$$

Proof. First, we see that

$$|\widetilde{e}_{ij} - f_{ij}| \leq \frac{(2\chi(\epsilon_F) + 21\epsilon_F)\|A\|\epsilon_F^2}{|\lambda_i - \lambda_j| - 14\|A\|\epsilon_F^2} \text{ for } i \neq j \text{ corresponding to } \lambda_i \neq \lambda_j$$

similar to the proof of (50) in Lemma 1. Concerning the multiple eigenvalues $\lambda_i = \lambda_j$, (45) yields $|\widetilde{\lambda}_i - \widetilde{\lambda}_j| \leq 14\|A\|\epsilon_F^2 < \delta$ from assumption (73). Recall that $\widetilde{e}_{ij} = \widetilde{e}_{ji} = r_{ij}/2$ whenever $|\widetilde{\lambda}_i - \widetilde{\lambda}_j| < \delta$ in Algorithm 1. Hence, from (62), we have

$$|\widetilde{e}_{ij} - f_{ij}| \leq \frac{\chi(\epsilon_F)}{2}\epsilon_F^2 \text{ for } i = j \text{ corresponding to } \lambda_i = \lambda_j. \tag{76}$$

Therefore, we have

$$\|F - \widetilde{E}\| \leq \sum_{1\leq i,j\leq n} \sqrt{|f_{ij} - \widetilde{e}_{ij}|^2} \leq \frac{(2\chi(\epsilon_F) + 21\epsilon_F)n\|A\|\epsilon_F^2}{\min_{\lambda_i\neq\lambda_j} |\lambda_i - \lambda_j| - 14\|A\|\epsilon_F^2} \tag{77}$$

similar to the proof of Lemma 1.

Next, we define

$$G := (X')^{-1}Y - I.$$

Then, we have

$$G = (I + \widetilde{E})^{-1}(F - \widetilde{E}) \tag{78}$$

similar to (58). Similar to the proof of (59), we have

$$\|G\| < \frac{5}{21}\|F\|$$

from (77) and (72). Using (65), we have

$$\|F'\| \leq 3\|G\|, \tag{79}$$

where $G := (X')^{-1}Y - I$ for $Y \in \{X\}$, $F' = (X')^{-1}Y' - I$ for $Y' \in \{X\}$, and the submatrices of $(X')^{-1}Y'$ corresponding to the multiple eigenvalues are symmetric positive definite. Therefore, we obtain (74). Since we see

$$\limsup_{\epsilon \to 0} \frac{\|G\|}{\|F\|^2} \leq \frac{6n\|A\|}{\min_{\lambda_i \neq \lambda_j} |\lambda_i - \lambda_j|}$$

from (78) and Lemma 3, we obtain (75) from (79). □

In the iterative refinement, Theorem 2 shows that the error term $\|F\|$ is quadratically convergent to zero, if the multiple eigenvalues can be identified by δ. Note that $\widehat{X}$ is also convergent to some fixed eigenvector matrix X, because Theorem 2 and (77) imply $\|\widetilde{E}\|/\|F\| \to 1$ as $\|F\| \to 0$ in $X' := \widehat{X}(I + \widetilde{E})$, where $\|F\|$ is quadratically convergent to zero.

In this paper, we cannot prove that δ satisfies (73) and cannot develop a strategy to mathematically achieve (73). However, we stress that $\delta = \mathcal{O}(\|F\|)$ is expected to be sufficiently larger than $\|F\|^2$, and (73) appears to be realized numerically for some appropriate ρ.

4.3 The complex case

For an Hermitian matrix $A \in \mathbb{C}^{n \times n}$, we must note that, for any unitary diagonal matrix U, XU is also an eigenvector matrix; there is a continuum of normalized eigenvector matrices, in contrast to the real case. Related to this, note that (25) is replaced with $\widetilde{E} + \widetilde{E}^{\mathrm{H}} = R$ in the complex case, and hence the diagonal elements $\widetilde{e}_{ii}$ for $i = 1, \ldots, n$ are not uniquely determined in $\mathbb{C}$. Now, choose $\widetilde{e}_{ii} = r_{ii}/2 \in \mathbb{R}$ for $i = 1, \ldots, n$. Then, we can prove quadratic convergence using the polar decomposition in the same way as in the discussion of multiple eigenvalues in the real case. More precisely, we define a normalized eigenvector matrix Y as follows. First, we focus on the situation where all the eigenvalues are simple. Define Y such that all the diagonal elements of $\widehat{X}^{-1}Y$ are positive real numbers. In addition, let $F := \widehat{X}^{-1}Y - I$. Then, we see quadratic convergence of F in the same way as in Theorem 2.

Corollary 1. *Let $A \in \mathbb{C}^{n\times n}$ be an Hermitian matrix whose eigenvalues are all simple. In addition, let $\{X\}$ be the set of the $n\times n$ unitary matrices whose columns are the eigenvectors of A. Suppose that Algorithm 1 is applied to A and $\widehat{X} \in \mathbb{R}^{n\times n}$ for some $\rho \geq 1$, and X' is obtained. Define $S = (s_{ij}) \in \mathbb{R}^{n\times n}$ such that $S = \widehat{X}^{\mathrm{H}} A \widehat{X}$, and*

$$\delta := \rho \cdot \max_{i\neq j} |s_{ij}|.$$

Let all the diagonal elements of $\widehat{X}^{-1}Y$ and $(X')^{-1}Y'$ be positive real numbers, where $Y, Y' \in \{X\}$. Furthermore, define F and F' such that $Y = \widehat{X}(I+F)$ and $Y' = X'(I+F')$, respectively. Suppose that

$$\epsilon_F := \|F\| < \frac{1}{3\rho} \min\left(\frac{\min_{\lambda_i\neq\lambda_j}|\lambda_i - \lambda_j|}{10n\|A\|}, \frac{1}{100}\right).$$

Then, we obtain

$$\|F'\| < \frac{5}{7}\|F\|,$$
$$\limsup_{\|F\|\to 0} \frac{\|F'\|}{\|F\|^2} \leq 3\left(\frac{6n\|A\|}{\min_{\lambda_i\neq\lambda_j}|\lambda_i - \lambda_j|}\right).$$

For a general Hermitian matrix, we define Y in the same manner as in Theorem 2, resulting in the following corollary.

Corollary 2. *Let $A \in \mathbb{C}^{n\times n}$ be an Hermitian matrix with multiple eigenvalues and the index sets $\mathcal{M}_k$, $k = 1, 2, \dots, M$ satisfy (63). In addition, let $\{X\}$ be the set of the $n\times n$ unitary matrices whose columns are the eigenvectors of A. Suppose that Algorithm 1 is applied to A and $\widehat{X} \in \mathbb{R}^{n\times n}$ for some $\rho \geq 1$, and X' is obtained. Define $S = (s_{ij}) \in \mathbb{R}^{n\times n}$ such that $S = \widehat{X}^{\mathrm{H}} A \widehat{X}$, and*

$$\delta := \rho \cdot \max_{i\neq j} |s_{ij}|.$$

In addition, define $Y, Y' \in \{X\}$ satisfying the following conditions. Let all the diagonal elements of $\widehat{X}^{-1}Y$ and $(X')^{-1}Y'$ be positive real numbers, and, for all k, the $n_k \times n_k$ submatrices of $\widehat{X}^{-1}Y$ and $(X')^{-1}Y'$ corresponding to $\{\lambda_i\}_{i\in\mathcal{M}_k}$ be symmetric and positive definite. Furthermore, define F and F' such that $Y = \widehat{X}(I+F)$ and $Y' = X'(I+F')$, respectively. Suppose that

$$\epsilon_F := \|F\| < \frac{1}{3\rho} \min\left(\frac{\min_{\lambda_i\neq\lambda_j}|\lambda_i - \lambda_j|}{10n\|A\|}, \frac{1}{100}\right).$$

In addition, assume that

$$14\|A\|\|F\|^2 < \delta.$$

Then, we obtain

$$\begin{aligned} &\|F'\| < \frac{5}{7}\|F\|, \\ &\limsup_{\|F\|\to 0} \frac{\|F'\|}{\|F\|^2} \le 3\left(\frac{6n\|A\|}{\min_{\lambda_i \neq \lambda_j} |\lambda_i - \lambda_j|}\right). \end{aligned}$$

In addition, note that $\widehat{X}$ is convergent to some fixed eigenvector matrix X in the same manner as the real case.

5 Algorithm for clustered eigenvalues

In this section, we propose the complete version of a refinement algorithm for the eigenvectors of symmetric matrices, which can also deal with clustered eigenvalues.

In the first place, we concretely show the drawback of Algorithm 1 concerning clustered eigenvalues. For this purpose, we again take the matrix A as in example (7) with $\varepsilon = 2^{-50} \approx 10^{-15}$. Recall that $\lambda_1 = -1$, $\lambda_2 = 2$, $\lambda_3 = 2+2\varepsilon$, i.e., λ_2 and λ_3 are nearly double eigenvalues. In the same way as before, we adopt the MATLAB built-in function `eig` in binary64 for obtaining $X^{(0)} := \widehat{X}$, which means $\mathbf{u} \approx 10^{-16}$ as the relative rounding error unit. Then, we iteratively apply Algorithm 1 to A and $X^{(\nu)}$ starting from $\nu = 0$. To check on the accuracy of $X^{(\nu)}$ with respect to the orthogonality and the diagonality, we display $R^{(\nu)} := I - (X^{(\nu)})^{\mathrm{T}}X^{(\nu)}$ and $S^{(\nu)} := (X^{(\nu)})^{\mathrm{T}}AX^{(\nu)}$. For $X^{(0)}$, we have the following results:

$$R^{(0)} \approx \begin{bmatrix} \texttt{-2.7e-16} & \texttt{-1.3e-16} & \texttt{-6.8e-17} \\ \texttt{-1.3e-16} & \texttt{1.4e-16} & \texttt{-5.0e-17} \\ \texttt{-6.8e-17} & \texttt{-5.0e-17} & \texttt{-2.2e-16} \end{bmatrix}, \quad S^{(0)} \approx \begin{bmatrix} \texttt{-1.0e+00} & \texttt{-1.3e-16} & \texttt{-6.8e-17} \\ \texttt{-1.3e-16} & \texttt{2.0e+00} & \texttt{1.7e-17} \\ \texttt{-6.8e-17} & \texttt{1.7e-17} & \texttt{2.0e+00} \end{bmatrix}$$

We apply Algorithm 1 to A and $X^{(0)}$ with $\rho = 1$ in the exact arithmetic. The result is as follows:

$$R^{(1)} \approx \begin{bmatrix} \texttt{5.4e-32} & \texttt{-3.2e-18} & \texttt{6.0e-18} \\ \texttt{-3.2e-18} & \texttt{-2.2e-03} & \texttt{-8.3e-18} \\ \texttt{6.0e-18} & \texttt{-8.3e-18} & \texttt{-2.2e-03} \end{bmatrix}, \quad S^{(1)} \approx \begin{bmatrix} \texttt{-1.0e+00} & \texttt{-3.2e-18} & \texttt{6.0e-18} \\ \texttt{-3.2e-18} & \texttt{2.0e+00} & \texttt{1.7e-17} \\ \texttt{6.0e-18} & \texttt{1.7e-17} & \texttt{2.0e+00} \end{bmatrix}$$

From the result, we confirm that the orthogonality of $X^{(\nu)}$, especially approximate eigenvectors corresponding to λ_2 and λ_3, is much worse than that to λ_1, and the diagonality is almost not improved. This result is consistent with the convergence analysis in the previous section because the error $\epsilon = \|E\|$ in Theorem 1 is not sufficiently small in this example though the radius of convergence in (55) is maximized at $\rho = 1$.

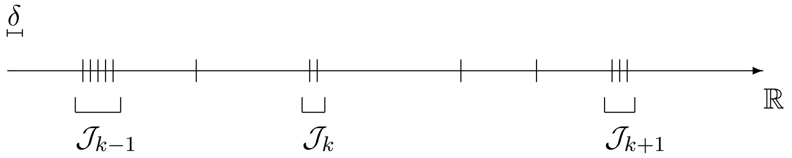

Figure 1: The relationship between $\widetilde{\lambda}_i$ and $\mathcal{J}_k$ (Short vertical lines denote $\widetilde{\lambda}_i$).

To solve the problem above, recall that the eigenspace spanned by all the eigenvectors corresponding to clustered eigenvalues is not sensitive to perturbations, as staed by the Davis–Kahan theorem in (4). In addition, noting the convergence analysis in Section 4.2 for multiple eigenvalues, we consider relaxing the criterion for judging whether A has clustered eigenvalues by setting ρ larger, in order to identify the eigenspace corresponding to the clustered eigenvalues. Heuristics suggest that ρ satisfying $10^2 \leq \rho \leq 10^{14}$ is a good choice (see the numerical results in Section 6). Following are the results of two iterations for $\rho = 10^3$:

$$R^{(1)} \approx \begin{bmatrix} \texttt{5.4e-32} & \texttt{2.7e-32} & \texttt{2.9e-32} \\ \texttt{2.7e-32} & \texttt{3.3e-32} & \texttt{1.2e-32} \\ \texttt{2.9e-32} & \texttt{1.2e-32} & \texttt{4.2e-32} \end{bmatrix}, \quad S^{(1)} \approx \left[\begin{array}{r|rr} \texttt{-1.0e+00} & \texttt{2.7e-32} & \texttt{2.9e-32} \\ \hline \texttt{2.7e-32} & \texttt{2.0e+00} & \texttt{-8.3e-17} \\ \texttt{2.9e-32} & \texttt{-8.3e-17} & \texttt{2.0e+00} \end{array}\right]$$

$$R^{(2)} \approx \begin{bmatrix} \texttt{2.2e-63} & \texttt{2.1e-63} & \texttt{2.3e-63} \\ \texttt{2.1e-63} & \texttt{1.7e-63} & \texttt{1.4e-63} \\ \texttt{2.3e-63} & \texttt{1.4e-63} & \texttt{2.3e-63} \end{bmatrix}, \quad S^{(2)} \approx \left[\begin{array}{r|rr} \texttt{-1.0e+00} & \texttt{2.1e-63} & \texttt{2.3e-63} \\ \hline \texttt{2.1e-63} & \texttt{2.0e+00} & \texttt{-8.3e-17} \\ \texttt{2.3e-63} & \texttt{-8.3e-17} & \texttt{2.0e+00} \end{array}\right]$$

We can confirm from $R^{(\nu)}$ that the orthogonality of $X^{(\nu)}$ is improved quadratically due to $\rho \gg 1$. Moreover, we see from $S^{(\nu)}$ that the diagonality corresponding to the simple eigenvalue λ_1 is also improved quadratically. On the other hand, the refinement of the diagonality stagnates with respect to the nearly double eigenvalues λ_2 and λ_3. In what follows, we overcome such a problem for a general symmetric matrix A.

Suppose that Algorithm 1 is applied to a symmetric matrix $A \in \mathbb{R}^{n \times n}$ and its approximate eigenvector matrix $\widehat{X} \in \mathbb{R}^{n \times n}$. Then, we obtain X', $\widetilde{\lambda}$ and δ, where $X' \in \mathbb{R}^{n \times n}$ is a refined approximate eigenvector matrix, and $\delta \in \mathbb{R}$ is the criterion whether approximate eigenvalues are clustered. Using $\widetilde{\lambda}$ and δ, we can easily obtain the index sets $\mathcal{J}_k$, $k = 1, 2, \dots, M$, for clusters of approximate eigenvalues $\{\widetilde{\lambda}_i\}_{i \in \mathcal{J}_k}$ satisfying all the following conditions (see also Fig. 1):

$$\begin{cases} \text{(a) } \mathcal{J}_k \subseteq \{1, 2, \dots, n\} \text{ with } n_k := |\mathcal{J}_k| \geq 2 \\ \text{(b) } \displaystyle\min_{j \in \mathcal{J}_k \setminus \{i\}} |\widetilde{\lambda}_i - \widetilde{\lambda}_j| \leq \delta, \ \forall i \in \mathcal{J}_k \\ \text{(c) } |\widetilde{\lambda}_i - \widetilde{\lambda}_j| > \delta, \ \forall i \in \mathcal{J}_k, \ \forall j \in \{1, 2, \dots, n\} \setminus \mathcal{J}_k \end{cases} . \tag{80}$$

Now the problem is how to refine $X'(:, \mathcal{J}_k) \in \mathbb{R}^{n \times n_k}$, which denotes the matrix comprising approximate eigenvectors corresponding to the clustered approximate eigenvalues $\{\widetilde{\lambda}_i\}_{i\in\mathcal{J}_k}$.

5.1 Outline of the proposed algorithm

From the observation on the numerical results at the beginning of this section, we develop the following procedure for the refinement:

1. Find clusters of approximate eigenvalues of A, and obtain the index sets $\mathcal{J}_k$, $k = 1, 2, \ldots, M$, for the clusters.
2. Define $V_k := X'(:, \mathcal{J}_k) \in \mathbb{R}^{n\times n_k}$ where $n_k := |\mathcal{J}_k|$.
3. Compute $T_k := V_k^{\mathrm{T}}(A - \mu_k I)V_k$ where $\mu_k := \widetilde{\lambda}_p$ with $|\widetilde{\lambda}_p| = \min_{i\in\mathcal{J}_k} |\widetilde{\lambda}_i|$.
4. Perform the following procedure for each T_k.
   i) Execute the eigenvalue decomposition of T_k such that $T_k = W_k D_k W_k^{\mathrm{T}}$ where W_k is the eigenvector matrix of T_k.
   ii) Update $X'(:, \mathcal{J}_k)$ by $V_k W_k$.

This procedure is interpreted as follows. We first apply an approximately similarity transformation to A using the refined eigenvector matrix X' such as $S := (X')^{\mathrm{T}}AX'$. Then, we divide the problem for S into subproblems for S_k, $k = 1, 2, \ldots, M$, corresponding to the clusters. After that, we apply a diagonal shift to S_k such as $T_k := S_k - \mu_k I$ for relatively separating the clustered eigenvalues around μ_k. Instead of these to obtain T_k, we actually perform the steps 2 and 3 in view of computational efficiency and accuracy. Finally, we update the columns of X' corresponding to $\mathcal{J}_k$ using the eigenvector matrix W_k of T_k by $V_k W_k$.

5.2 The proposed algorithm

In Algorithm 2, we present the complete version of a refinement algorithm for the eigenvalue decomposition of a real symmetric matrix A, which can also be applied to the case where A has clustered eigenvalues.

Here, the function $\mathsf{FL}(C)$ rounds an input matrix $C \in \mathbb{R}^{n\times n}$ to the matrix $T \in \mathbb{F}^{n\times n}$ nearest to C, where $\mathbb{F}$ is a set of ordinary floating-point numbers such as IEEE 754 binary64 format. Moreover, the function $\mathsf{eig}(T)$ is similar to the MATLAB's one, which computes all approximate eigenvectors of an input matrix $T \in \mathbb{F}^{n\times n}$ by using ordinary floating-point arithmetic, and is expected to adopt some backward stable algorithm as implemented in the LAPACK routine xSYEV. To obtain sufficiently accurate approximate eigenvectors of A corresponding to $\mathcal{J}_k$, we iteratively apply Algorithm 1 (RefSyEv) to $A - \mu_k I$ and $V_k^{(\nu)}$ until the approximate eigenvectors as the

Algorithm 2 RefSyEvCL: Refinement of approximate eigenvectors of a real symmetric matrix with clustered eigenvalues.

```
Require: $A, \widehat{X} \in \mathbb{R}^{n\times n}$ with $A = A^{\mathrm{T}}$;  $\rho \in \mathbb{R}$ with $\rho \geq 1$
Ensure: $X' \in \mathbb{R}^{n\times n}$
 1: function $X' \leftarrow$ RefSyEvCL($A$, $\widehat{X}$, $\rho$)
 2:     $[X', e_{\max}, \widetilde{\lambda}, \delta] \leftarrow$ RefSyEv($A, \widehat{X}, \rho$)    ▷ Apply Algorithm 1 to $A$ and $\widehat{X}$.
 3:     if $e_{\max} \geq 1$, return, end if    ▷ Improvement cannot be expected.
 4:     Determine the index sets $\mathcal{J}_k$, $k = 1, \dots, M$, as in (80) for eigenvalue clusters using $\widetilde{\lambda}$ and $\delta$.    ▷ $M$: The number of clusters.
 5:     for $k \leftarrow 1, 2, \dots, M$ do    ▷ Refine eigenvectors for each cluster.
 6:         $V_k \leftarrow X'(:, \mathcal{J}_k)$    ▷ Pick out $V_k \in \mathbb{R}^{n\times n_k}$ where $n_k := |\mathcal{J}_k|$.
 7:         $\mu_k \leftarrow \widetilde{\lambda}_p$ s.t. $|\widetilde{\lambda}_p| = \min_{i\in\mathcal{J}_k} |\widetilde{\lambda}_i|$    ▷ Determine the shift constant $\mu_k$.
 8:         $A_k \leftarrow A - \mu_k I$    ▷ Shift $A$ for separating clustered eigenvalues.
 9:         $T_k \leftarrow \mathsf{FL}(V_k^{\mathrm{T}} A_k V_k)$    ▷ Round $V_k^{\mathrm{T}} A_k V_k$ to floating-point.
10:         $[W_k, \sim] \leftarrow \mathsf{eig}(T_k)$    ▷ Compute eigenvectors of $T_k$ in floating-point.
11:         $\nu \leftarrow 1$;  $V_k^{(1)} \leftarrow V_k \cdot W_k$
12:         repeat
13:             $[V_k^{(\nu+1)}, f_{\max}] \leftarrow$ RefSyEv($A_k, V_k^{(\nu)}, \rho$)    ▷ Apply Alg. 1 to $A_k$ and $V_k^{(\nu)}$.
14:             if $f_{\max} \geq 1$, return, end if    ▷ Improvement cannot be expected.
15:             $\nu \leftarrow \nu + 1$
16:         until $f_{\max} \leq e_{\max}$
17:         $X'(:, \mathcal{J}_k) \leftarrow V_k^{(\nu)}$    ▷ Update $X'$.
18:     end for
19: end function
```

columns of $V_k^{(\nu)}$ become as accurate as those associated with well-separated eigenvalues, which corresponds to the lines from 12 to 16 in Algorithm 2.

For the example (7), we apply Algorithm 2 (RefSyEvCL) to A and the same initial guess $X^{(0)}$ as before. Following are the results of two iterations for $\rho = 10^3$:

$$R^{(1)} \approx \begin{bmatrix} \texttt{5.4e-32} & \texttt{-2.9e-32} & \texttt{2.8e-32} \\ \texttt{-2.9e-32} & \texttt{1.4e-33} & \texttt{7.9e-49} \\ \texttt{2.8e-32} & \texttt{7.9e-49} & \texttt{1.4e-33} \end{bmatrix}, \quad S^{(1)} \approx \begin{bmatrix} \texttt{-1.0e+00} & \texttt{-2.9e-32} & \texttt{2.8e-32} \\ \texttt{-2.9e-32} & \texttt{2.0e+00} & \texttt{-1.0e-48} \\ \texttt{2.8e-32} & \texttt{-1.0e-48} & \texttt{2.0e+00} \end{bmatrix}$$

$$R^{(2)} \approx \begin{bmatrix} \texttt{2.2e-63} & \texttt{-1.6e-63} & \texttt{1.5e-63} \\ \texttt{-1.6e-63} & \texttt{8.1e-64} & \texttt{-7.9e-64} \\ \texttt{1.5e-63} & \texttt{-7.9e-64} & \texttt{7.6e-64} \end{bmatrix}, \quad S^{(2)} \approx \begin{bmatrix} \texttt{-1.0e+00} & \texttt{-1.6e-63} & \texttt{1.5e-63} \\ \texttt{-1.6e-63} & \texttt{2.0e+00} & \texttt{-7.9e-64} \\ \texttt{1.5e-63} & \texttt{-7.9e-64} & \texttt{2.0e+00} \end{bmatrix}$$

From the results, we can see that Algorithm 2 works well for this example, i.e., the approximate eigenvectors corresponding to the nearly double eigenvalues λ_2 and λ_3 are also improved in terms of the orthogonality and the diagonality.

6 Numerical results

We present several numerical results to illustrate the effectiveness of the proposed algorithms (Algorithms 1 and 2) for symmetric eigenvalue decomposition. Numerical experiments in this section were conducted using MATLAB R2015b on our PC with 2.9 GHz Intel Xeon CPU E5-4617 (6 cores × 4 CPUs) and 1 TB of main memory. We adopt IEEE 754 binary64 (formerly double precision) as the working precision of floating-point arithmetic.

6.1 Convergence property

We confirm the convergence property of the proposed algorithms. For this purpose, we first generate real symmetric positive definite matrices with various distributions of eigenvalues using MATLAB's built-in function randsvd from Higham's test matrices [15] by the following MATLAB command:

```
>> A = gallery('randsvd',n,-cnd,mode);
```

The eigenvalue distribution and the condition number of A can be controlled by the input arguments `mode` $\in \{1, 2, 3, 4, 5\}$ and `cnd` $=: \alpha \geq 1$, as follows:

1. one large: $\lambda_1 \approx 1$, $\lambda_i \approx \alpha^{-1}$, $i = 2, \ldots, n$
2. one small: $\lambda_n \approx \alpha^{-1}$, $\lambda_i \approx 1$, $i = 1, \ldots, n-1$
3. geometrically distributed: $\lambda_i \approx \alpha^{-(i-1)/(n-1)}$, $i = 1, \ldots, n$
4. arithmetically distributed: $\lambda_i \approx 1-(1-\alpha^{-1})(i-1)/(n-1)$, $i = 1, \ldots, n$
5. random with uniformly distributed logarithm: $\lambda_i \approx \alpha^{-r(i)}$, $i = 1, \ldots, n$, where $r(i)$ are pseudo-random values drawn from the standard uniform distribution on $(0, 1)$.

Here, $\kappa(A) \approx$ `cnd` for `cnd` $< \mathbf{u}^{-1} \approx 10^{16}$. Note that for `mode` $\in \{1, 2\}$ there are clustered eigenvalues, which are not exactly but nearly multiple eigenvalues as a result of rounding errors when A is generated using randsvd, i.e., all the multiple eigenvalues are slightly separated by the perturbation. Therefore, we expect that Algorithm 1 (RefSyEv) does not work effectively for `mode` $\in \{1, 2\}$.

We set $n = 10$ and `cnd` $= 10^8$. Here, we performed numerical experiments for some dozens of seeds for the random number generator, and all the results were similar to those provided in this section. Therefore, we adopt the default seed as a typical example by the MATLAB command

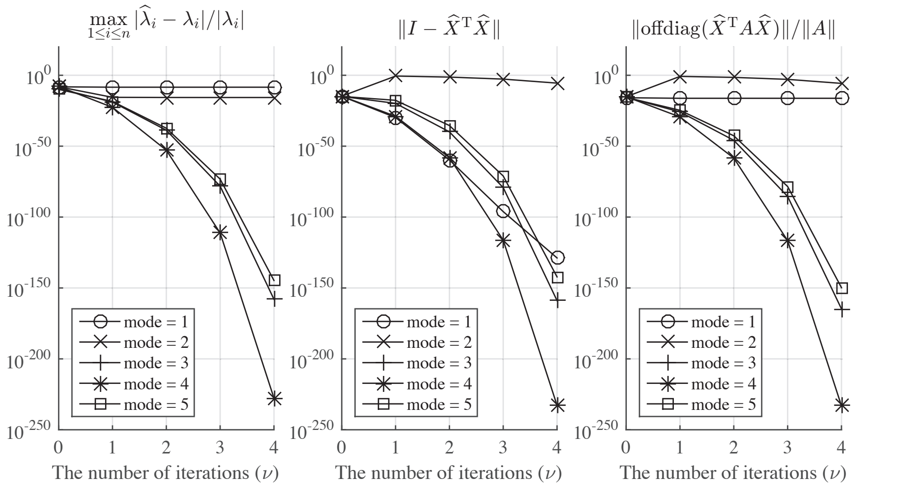

Figure 2: Results of iterative refinement by Algorithm 1 (RefSyEv, $\rho = 1$) for symmetric and positive definite matrices generated by `randsvd` with $n = 10$ and $\kappa(A) \approx 10^8$.

`rng('default')` to ensure the reproducibility. Moreover, we compute $\widehat{X}^{(0)}$ as an initial approximate eigenvector matrix using the MATLAB function `eig` for the eigenvalue decomposition in IEEE binary64 floating-point arithmetic. Therefore, $\widehat{X}^{(0)}$ suffers from rounding errors. To see the behavior of Algorithm 1 exactly, in Algorithm 1 we use the Symbolic Math Toolbox for MATLAB. In addition, we set $\rho = 1$ in Algorithm 1. The results are displayed in Fig. 2, which provides $\max_{1\le i\le n} |\widehat{\lambda}_i - \lambda_i|/|\lambda_i|$ as the maximum relative error of computed eigenvalues (left), $\|I - \widehat{X}^{\mathrm{T}}\widehat{X}\|$ as the orthogonality of a computed eigenvector matrix $\widehat{X}$ (center) and $\|\mathrm{offdiag}(\widehat{X}^{\mathrm{T}}A\widehat{X})\|/\|A\|$ as the diagonality of $\widehat{X}^{\mathrm{T}}A\widehat{X}$ (right). Here, offdiag$(\cdot)$ denotes the off-diagonal part. The horizontal axis shows the number of iterations of Algorithm 1.

In the case of `mode` $\in \{3, 4, 5\}$, Algorithm 1 converges quadratically, as expected. On the other hand, in the case of `mode` $\in \{1, 2\}$, the algorithm fails to improve the accuracy of initial approximate eigenvectors. This is because the test matrices for `mode` $\in \{1, 2\}$ have nearly multiple eigenvalues, and the assumption (55) for the convergence of Algorithm 1 is not satisfied for these eigenvalues.

To confirm the behavior of Algorithm 2 (RefSyEvCL), we apply it to the same examples with various ρ. The results are displayed in Fig. 3, whose horizontal axis shows the number of iterations of Algorithm 2. It can be seen from the results that Algorithm 2 also works very well for $10^2 \le \rho \le 10^{14}$, even in the case of `mode` $\in \{1, 2\}$, indicating quadratic convergence.

We mention that Algorithm 1 works perfectly for exactly multiple eigenvalues. Let $A = I + ee^{\mathrm{T}} \in \mathbb{R}^{n\times n}$ with $e = (1, \ldots, 1)^{\mathrm{T}}$. Then A is exactly

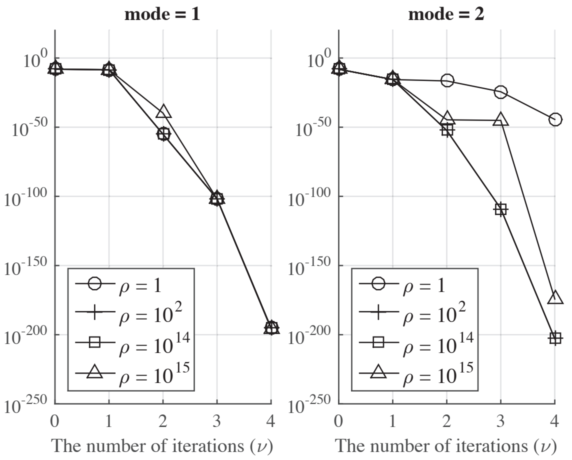

Figure 3: Maximum relative error of computed eigenvalues refined by Algorithm 2 (RefSyEvCL) with various ρ for symmetric and positive definite matrices generated by randsvd with $n = 10$, $\kappa(A) \approx 10^8$, mode $\in \{1, 2\}$.

representable in the IEEE 754 binary64 floating-point format and has exactly $(n-1)$-fold eigenvalues ($\lambda_i = 1$, $i = 1, \dots, n-1$, $\lambda_n = n+1$). In the same way as in the above examples, we apply Algorithm 1 for A with $n = 10$. The results are displayed in Fig. 4.

From the results, we can see that Algorithm 1 also converges quadratically for the matrix having exactly multiple eigenvalues, which is consistent with our convergence analysis in Theorem 2.

Second, we show the results for the Wilkinson matrix [26] with $n = 21$, which is symmetric and tridiagonal with pairs of nearly but not exactly equal eigenvalues. The Wilkinson matrix $W_n = (w_{ij}) \in \mathbb{R}^{n\times n}$ consists of diagonal entries $w_{ii} := \frac{|n-2i+1|}{2}$, $i = 1, 2, \dots, n$, and off-diagonal entries being all ones. We apply Algorithm 2 with various ρ to the Wilkinson matrix. The results are displayed in Fig. 5. From the results, we can see that Algorithm 2 works well in the case where $10^2 \le \rho \le 10^{14}$, especially it does very well if appropriate ρ is chosen such as $\rho = 10^{14}$.

Third, we show the convergence behavior of Algorithm 1 with limited computational precision for larger matrices with various condition numbers. For this purpose, we generate test matrices by again using randsvd with n $= 100$, and mode $= 3$ and varying cnd from 10^3 to 10^{15}. In the same way as in the previous examples, we use eig in binary64 for calculating a matrix of initial approximate eigenvectors. Moreover, we represent $\widehat{X} = \widehat{X}_1 + \widehat{X}_2$ in "double-double" precision format [8] with the leading part $\widehat{X}_1$ and the trailing part $\widehat{X}_2$ to simulate twice the working precision by using the concept of error-free transformations [18, 20, 21, 23, 24]. Then the maximum relative

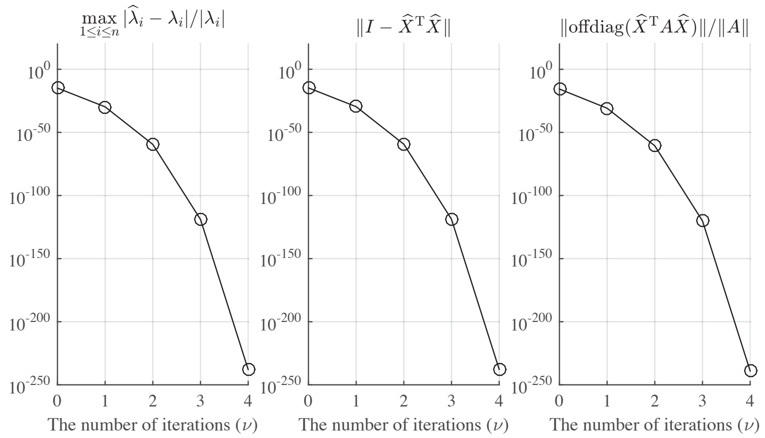

Figure 4: Results of iterative refinement by Algorithm 1 (RefSyEv, $\rho = 1$) for a symmetric 10×10 matrix $A = I + ee^{\mathrm{T}}$ with $e = (1, \ldots, 1)^{\mathrm{T}}$ having exactly 9-fold eigenvalues.

accuracy of the computed results is limited to $\mathbf{u}_h = \mathbf{u}^2$, that is approximately 10^{-32} in this case, since $\mathbf{u} = 2^{-53} \approx 10^{-16}$ for binary64. Here, we set $\rho = 1$ in Algorithm 1. The results are displayed in Fig. 6.

From the results in Fig. 6, we see that the quadratic convergence of Algorithm 1 can be confirmed until the relative accuracy of the computed results (`MaxRelErr`), the orthogonality of the computed eigenvectors ($\|R\|$), and their relative diagonality ($\|S_{\mathrm{off}}\|/\|A\|$) attain approximately $\mathbf{u}_h = \mathbf{u}^2 \approx 10^{-32}$, except for the case `cnd` $= 10^{15}$ ($\kappa(A) \approx 10^{15}$). This result is consistent with the discussion in Remark 4. In the case of `cnd` $= 10^{15}$, Algorithm 1 does not work effectively due to the ill-conditionedness of A, i.e., smaller magnitude eigenvalues of A are regarded as nearly multiple, compared to the largest magnitude eigenvalue of A. This problem can be resolved by Algorithm 2. The results obtained by Algorithm 2 are displayed in Fig. 7, which shows that Algorithm 2 can effectively improve eigenvectors even if A is ill-conditioned.

6.2 Computational speed

To evaluate the computational speed of Algorithm 1, we first compare computing time for Algorithm 1 with that for the "MP-approach" which means a native approach using multiple-precision arithmetic. Note that the timing should be observed for reference because the computing time for Algorithm 1 strongly depends on the implementation of accurate matrix multiplication. For this purpose, we adopt an efficient method by Ozaki et al. [21], which can utilize fast matrix multiplication routines such as xGEMM in Basic Linear

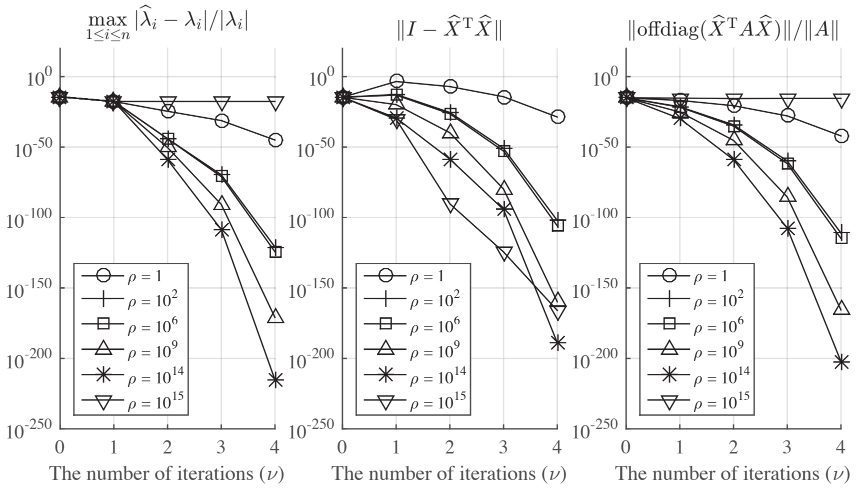

Figure 5: Results of iterative refinement by Algorithm 2 (RefSyEvCL) with various ρ for the Wilkinson matrix with $n = 21$.

Algebra Subprograms (BLAS). As an MP-approach, we use Advanpix Multiprecision Computing Toolbox version 3.8.5.9059 [2], which utilizes well-known, fast, and reliable multiple-precision arithmetic libraries including GMP and MPFR. In the toolbox, the MRRR algorithm [10] via Householder's reduction is used for solving symmetric eigenvalue problems.

As test matrices, we generate pseudo-random real symmetric $n \times n$ matrices with $n \in \{500, 1000\}$ using the MATLAB function `randn` such as `B = randn(n)` and `A = B + B'`. We use `eig` in binary64, which adopts DSYEV in LAPACK, and then iteratively refine the computed eigenvalues and eigenvectors three times. Here, we set $\rho = 1$ in Algorithm 1. In the multiple-precision toolbox, we can control the arithmetic precision `d` in decimal digits using the command `mp.Digits(d)`. In particular, IEEE 754 binary128 arithmetic is supported as a special case for $\mathtt{d} = 34$, which is faster than the cases for $\mathtt{d} < 34$. Corresponding to the results of Algorithm 1, we adjust `d` for $\nu = 1, 2, 3$. For timing fairness, we adopt $\mathtt{d} = \max(\mathtt{d}, 34)$. In Tables 1 and 2, we display the maximum relative error of computed eigenvalues $\widehat{\lambda}_i$ and the measured computing time.

From the tables, it can be seen that Algorithm 1 quadratically improves the accuracy of the computed eigenvalues. The accuracy of the results obtained using the MP-approach corresponds to the arithmetic precision `d`. Note that on the timing for Algorithm 1, the symmetry of $\widehat{X}^{\mathrm{T}}\widehat{X}$ or $\widehat{X}^{\mathrm{T}}A\widehat{X}$ was not considered for their computations. Nevertheless, Algorithm 1 is considerably faster than the MP-approach, especially for larger n. For larger ν, there must be some cross point at which the MP-approach outperforms the proposed method in terms of computing time, though it may be out of

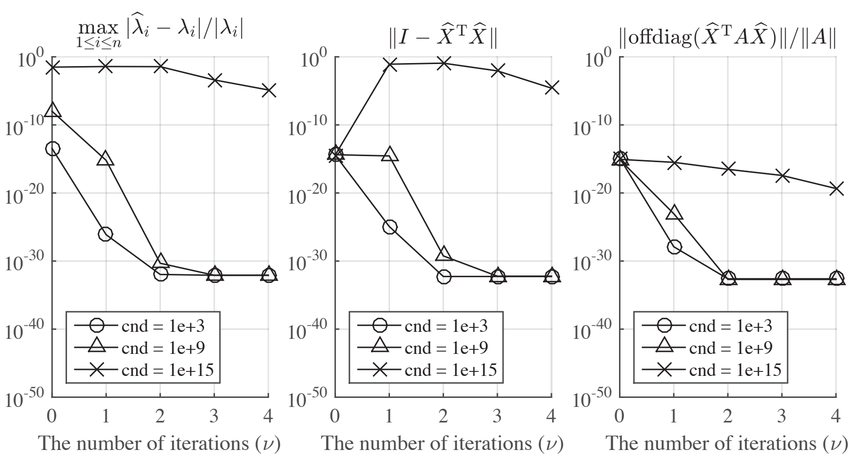

Figure 6: Results of iterative refinement by Algorithm 1 (RefSyEv) with $\rho = 1$ using double-double precision format for symmetric positive definite matrices generated by randsvd with $\mathtt{n} = 100$, $\mathtt{mode} = 3$ and various condition numbers.

range of the floating-point numbers and arithmetic being used.

We deal with more large-scale problems. We aim to obtain maximally accurate all eigenpairs of a given real symmetric $n \times n$ matrix A in binary64 format. For this purpose, we apply the MATLAB function eig in binary64 to A for calculating its initial approximate eigenvector matrix $\widehat{X}$, and then refine $\widehat{X}$ by Algorithm 1 with double-double precision format as mentioned in the previous section. After that, we round the computed results back to binary64 format. As a matrix multiplication routine in double-double precision format, we adopt the method in [21] again.

Test matrices are generated by using the MATLAB function randsvd with $n \in \{4000, 8000, 16000\}$, $\mathtt{cnd} = 10^8$ and $\mathtt{mode} = 3$. As numerical results, the following items are provided:

- MaxRelErr: $\displaystyle\max_{1\le i\le n} \frac{|\widehat{\lambda}_i - \lambda_i|}{|\lambda_i|}$ (on accuracy of eigenvalues)
- $\|R\|$: $\|I - \widehat{X}^{\mathrm{T}}\widehat{X}\|$ (on orthogonality of eigevectors)
- $\|S_{\mathrm{off}}\|/\|A\|$: $\displaystyle\frac{\|\mathrm{offdiag}(\widehat{X}^{\mathrm{T}}A\widehat{X})\|}{\|A\|}$ (on diagonality of eigevectors)

The results are displayed in Table 3.

From Table 3, it can be seen that Algorithm 1 improves the accuracy of the computed results up to the limit of binary64 ($\mathbf{u} = 2^{-53} \approx 10^{-16}$). Computing time for the proposed refinement algorithm is comparable to that

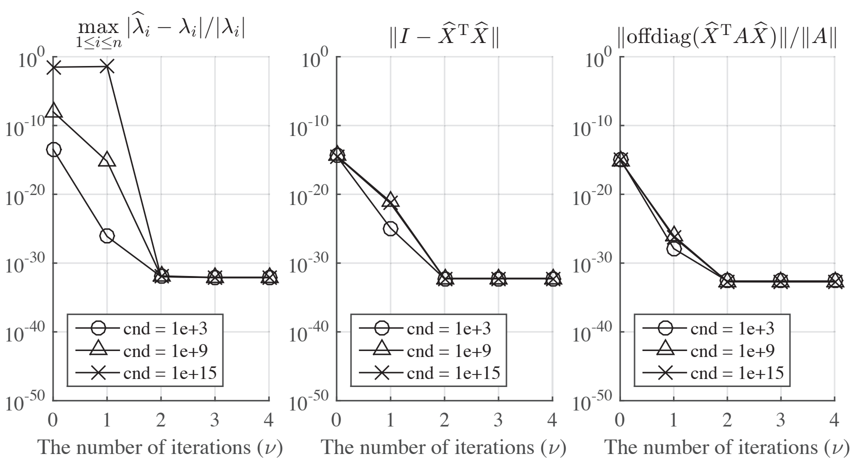

Figure 7: Results of iterative refinement by Algorithm 2 (RefSyEvCL) with $\rho = 10^9$ using double-double precision format for symmetric positive definite matrices generated by randsvd with $\mathtt{n} = 100$, $\mathtt{mode} = 3$ and various condition numbers.

for eig in binary64, especially for larger n. It turns out that the proposed algorithms become useful for large-scale problems in practice if fast routines of accurate matrix multiplication are available.

7 Conclusion

We proposed novel refinement algorithms for the eigenvalue decomposition of real symmetric matrices, which can iteratively be applied. Quadratic convergence of the basic algorithm (Algorithm 1) was proved for well-separated eigenvalues as well as multiple ones in the same manner as in Newton's method. The complete version of the refinement algorithm (Algorithm 2) can improve approximate eigenvectors corresponding to not only well-separated eigenvalues but also clustered ones. As shown theoretically and numerically, Algorithm 2 works well, provided that backward stable algorithms for the eigenvalue decomposition are available in ordinary floating-point arithmetic.

The proposed algorithms benefit from the availability of high efficiency matrix multiplication in higher-precision arithmetic in practice. Numerical results showed excellent performance of the proposed algorithms in terms of convergence rate and measured computing time. The accuracy of the results could be improved up to the limit of computational precision in use.

Acknowledgement
This study was partially supported by CREST, JST, and JSPS KAKENHI Grant Numbers 16H03917, 25790096.

Table 1: Results for a pseudo-random real symmetric matrix, $n = 500$.

Proposed algorithm	`eig` (binary64)	$\nu = 1$	$\nu = 2$	$\nu = 3$
$\max_i \lvert\widehat{\lambda}_i - \lambda_i\rvert/\lvert\lambda_i\rvert$	2.2×10^{-14}	3.8×10^{-28}	2.2×10^{-55}	1.9×10^{-107}
Elapsed Time (s)	0.09	2.23	4.42	12.69
(accumulated)		2.32	6.74	19.43
MP-approach	`mp.Digits(d)`	`d` = 34	`d` = 58	`d` = 109
$\max_i \lvert\widehat{\lambda}_i - \lambda_i\rvert/\lvert\lambda_i\rvert$		2.6×10^{-32}	1.2×10^{-55}	1.6×10^{-106}
Elapsed Time (s)		7.35	74.10	85.04

Table 2: Results for a pseudo-random real symmetric matrix, $n = 1000$.

Proposed algorithm	`eig` (binary64)	$\nu = 1$	$\nu = 2$	$\nu = 3$
$\max_i \lvert\widehat{\lambda}_i - \lambda_i\rvert/\lvert\lambda_i\rvert$	1.5×10^{-14}	1.3×10^{-27}	5.9×10^{-53}	7.3×10^{-102}
Elapsed Time (s)	0.22	11.19	29.05	83.19
(accumulated)		11.41	40.46	123.65
MP-approach	`mp.Digits(d)`	`d` = 34	`d` = 56	`d` = 105
$\max_i \lvert\widehat{\lambda}_i - \lambda_i\rvert/\lvert\lambda_i\rvert$		1.3×10^{-31}	6.0×10^{-53}	2.0×10^{-102}
Elapsed Time (s)		50.18	538.82	646.33

References

[1] P. A. Absil, R. Mahony, R. Sepulchre, P. Van Dooren, *A Grassmann-Rayleigh quotient iteration for computing invariant subspaces*, SIAM Rev., 44:1 (2006), pp. 57–73.

[2] Advanpix, *Multiprecision Computing Toolbox for MATLAB*, 2015. Code and documentation available at `http://www.advanpix.com/`.

[3] M. Ahuesa, A. Largillier, F. D. d'Almeida, P. B. Vasconcelos, *Spectral refinement on quasi-diagonal matrices*, Linear Algebra Appl., 401 (2005), pp. 109–117.

[4] A. R. Collar, *Some notes on Jahn's method for the improvement of approximate latent roots and vectors of a square matrix*, Quart. J. Mech. Appl. Math., 1 (1948), pp. 145–148.

[5] P. I. Davies, N. J. Higham, F. Tisseur, *Analysis of the Cholesky method with iterative refinement for solving the symmetric definite generalized eigenproblem*, SIAM. J. Matrix Anal. Appl., 23:2 (2001), pp. 472–493.

Table 3: Results of iterative refinement for symmetric and positive definite matrices generated by randsvd with cnd $= 10^8$ and mode $= 3$.

$n = 4000$	eig (binary64)	$\nu = 1$	$\nu = 2$
MaxRelErr	5.5×10^{-10}	2.7×10^{-16}	2.4×10^{-16}
$\Vert R \Vert$	1.7×10^{-14}	1.5×10^{-14}	1.3×10^{-16}
$\Vert S_{\text{off}} \Vert / \Vert A \Vert$	3.8×10^{-15}	5.3×10^{-17}	5.3×10^{-17}
Elapsed Time (s)	3.57	8.19	8.36
(accumulated)		11.76	20.11
$n = 8000$	eig (binary64)	$\nu = 1$	$\nu = 2$
MaxRelErr	6.2×10^{-10}	3.0×10^{-16}	2.6×10^{-16}
$\Vert R \Vert$	2.7×10^{-14}	6.3×10^{-14}	1.3×10^{-16}
$\Vert S_{\text{off}} \Vert / \Vert A \Vert$	4.5×10^{-15}	5.4×10^{-17}	5.4×10^{-17}
Elapsed Time (s)	64.95	52.66	51.72
(accumulated)		117.61	169.33
$n = 16000$	eig (binary64)	$\nu = 1$	$\nu = 2$
MaxRelErr	8.1×10^{-10}	3.5×10^{-16}	2.8×10^{-16}
$\Vert R \Vert$	1.1×10^{-13}	3.5×10^{-13}	1.5×10^{-16}
$\Vert S_{\text{off}} \Vert / \Vert A \Vert$	6.0×10^{-15}	5.4×10^{-17}	5.4×10^{-17}
Elapsed Time (s)	599.85	370.30	369.19
(accumulated)		970.15	1339.34

[6] R. O. Davies, J. J. Modi, *A direct method for completing eigenproblem solutions on a parallel computer*, Linear Algebra Appl., 77 (1986), pp. 61–74.

[7] P. I. Davies, M. I. Smith, *Updating the singular value decomposition*, J. Comput. Appl. Math., 170 (2004), pp. 145–167.

[8] *DDFUN90: Fortran-90 double-double package*, 2005. Code and documentation available at http://crd-legacy.lbl.gov/~dhbailey/.

[9] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[10] I. S. Dhillon, B. N. Parlett, *Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices*, Linear Algebra Appl., 387 (2004), pp. 1–28.

[11] J. J. Dongarra, C. B. Moler, J. H. Wilkinson, *Improving the accuracy of computed eigenvalues and eigenvectors*, SIAM J. Numer. Anal., 20:1 (1983), pp. 23–45.

[12] *GMP: GNU Multiple Precision Arithmetic Library*, 2015. Code and documentation available at `http://gmplib.org/`.

[13] G. H. Golub, C. F. Van Loan, *Matrix Computations*, 4th ed., The Johns Hopkins University Press, Baltimore, 2013.

[14] M. Gu, S. C. Eisenstat, *A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem*, SIAM J. Matrix Anal. Appl., 16:1 (1995), pp. 172–191.

[15] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, PA, 2002.

[16] N. J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, PA, 2008.

[17] H. A. Jahn, *Improvement of an approximate set of latent roots and modal columns of a matrix by methods akin to those of classical perturbation theory*, Quart. J. Mech. Appl. Math., 1 (1948), pp. 131–144.

[18] X. S. Li, J. W. Demmel, D. H. Bailey, G. Henry, Y. Hida, J. Iskandar, W. Kahan, S. Y. Kang, A. Kapur, M. C. Martin, B. J. Thompson, T. Tung, and D. Yoo, *Design, implementation and testing of extended and mixed precision BLAS*, ACM Trans. Math. Software, 28 (2002), pp. 152–205.

[19] *MPFR: The GNU MPFR Library*, 2013. Code and documentation available at `http://www.mpfr.org/`.

[20] T. Ogita, S. M. Rump, S. Oishi, *Accurate sum and dot product*, SIAM J. Sci. Comput., 26:6 (2005), pp. 1955–1988.

[21] K. Ozaki, T. Ogita, S. Oishi, S. M. Rump, *Error-free transformations of matrix multiplication by using fast routines of matrix multiplication and its applications*, Numerical Algorithms, 59:1 (2012), pp. 95-118.

[22] B. N. Parlett, *The Symmetric Eigenvalue Problem*, 2nd ed., Classics in Applied Mathematics, Vol. 20, SIAM, Philadelphia, 1998.

[23] S. M. Rump, T. Ogita, S. Oishi, *Accurate floating-point summation part I: faithful rounding*, SIAM J. Sci. Comput., 31:1 (2008), pp. 189–224.

[24] S. M. Rump, T. Ogita, S. Oishi, *Accurate floating-point summation part II: Sign, K-fold faithful and rounding to nearest*, SIAM J. Sci. Comput., 31:2 (2008), pp. 1269–1302.

[25] F. Tisseur, *Newton's method in floating point arithmetic and iterative refinement of generalized eigenvalue problems*, SIAM J. Matrix Anal. Appl., 22:4 (2001), pp. 1038–1057.

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