Analysis of the Cholesky Method with Iterative Refinement for Solving the Symmetric Definite Generalized Eigenproblem

Philip I. Davies, Nicholas J. Higham and
Françoise Tisseur

2001

MIMS EPrint: **2008.70**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://www.manchester.ac.uk/mims/eprints
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 472–493

ANALYSIS OF THE CHOLESKY METHOD WITH ITERATIVE REFINEMENT FOR SOLVING THE SYMMETRIC DEFINITE GENERALIZED EIGENPROBLEM*

PHILIP I. DAVIES†, NICHOLAS J. HIGHAM†, AND FRANÇOISE TISSEUR†

Abstract. A standard method for solving the symmetric definite generalized eigenvalue problem $Ax = \lambda Bx$, where A is symmetric and B is symmetric positive definite, is to compute a Cholesky factorization $B = LL^T$ (optionally with complete pivoting) and solve the equivalent standard symmetric eigenvalue problem $Cy = \lambda y$, where $C = L^{-1}AL^{-T}$. Provided that a stable eigensolver is used, standard error analysis says that the computed eigenvalues are exact for $A + \Delta A$ and $B + \Delta B$ with $\max(\|\Delta A\|_2/\|A\|_2, \|\Delta B\|_2/\|B\|_2)$ bounded by a multiple of $\kappa_2(B)u$, where u is the unit roundoff. We take the Jacobi method as the eigensolver and give a detailed error analysis that yields backward error bounds potentially much smaller than $\kappa_2(B)u$. To show the practical utility of our bounds we describe a vibration problem from structural engineering in which B is ill conditioned yet the error bounds are small. We show how, in cases of instability, iterative refinement based on Newton's method can be used to produce eigenpairs with small backward errors. Our analysis and experiments also give insight into the popular Cholesky–QR method, in which the QR method is used as the eigensolver. We argue that it is desirable to augment current implementations of this method with pivoting in the Cholesky factorization.

Key words. symmetric definite generalized eigenvalue problem, Cholesky method, Cholesky factorization with complete pivoting, Jacobi method, backward error analysis, rounding error analysis, iterative refinement, Newton's method, LAPACK, MATLAB

AMS subject classification. 65F15

PII. S0895479800373498

1. Introduction.

The symmetric definite generalized eigenvalue problem

$$Ax = \lambda Bx, \tag{1.1}$$

where $A, B \in \mathbb{R}^{n\times n}$ are symmetric and B is positive definite, arises in many applications in science and engineering [4, chapter 9], [16]. An important open problem is to derive a method of solution that takes advantage of the structure and is efficient and backward stable. Such a method should, for example, require half the storage of a method for the generalized nonsymmetric problem and produce real computed eigenvalues.

The QZ algorithm [18] can be used to solve (1.1). It computes orthogonal matrices Q and Z such that Q^TAZ is upper quasi-triangular and Q^TBZ is upper triangular. This method is numerically stable but it does not exploit the special structure of the problem and so does not necessarily produce real eigenpairs in floating point arithmetic.

*Received by the editors June 9, 2000; accepted for publication (in revised form) by M. Chu April 11, 2001; published electronically September 7, 2001.

http://www.siam.org/journals/simax/23-2/37349.html

†Department of Mathematics, University of Manchester, Manchester, M13 9PL, England (ieuan@ma.man.ac.uk, http://www.ma.man.ac.uk/~ieuan/, higham@ma.man.ac.uk, http://www.ma.man.ac.uk/~higham/, ftisseur@ma.man.ac.uk, http://www.ma.man.ac.uk/~ftisseur/). The work of the first author was supported by an Engineering and Physical Sciences Research Council CASE Ph.D. Studentship with NAG Ltd. (Oxford) as the cooperating body. The work of the second author was supported by Engineering and Physical Sciences Research Council grant GR/L76532 and a Royal Society Leverhulme Trust Senior Research Fellowship. The work of the third author was supported by Engineering and Physical Sciences Research Council grant GR/L76532.

A method that potentially has the desired properties has recently been proposed by Chandrasekaran [3], but the worst-case computational cost of this algorithm is not clear.

A standard method, apparently first suggested by Wilkinson [25, pp. 337–340], begins by computing the Cholesky factorization, optionally with complete pivoting [12, section 4.2.9], [14, section 10.3],

$$\Pi^T B \Pi = L D^2 L^T, \tag{1.2}$$

where Π is a permutation matrix, L is unit lower triangular, and $D^2 = \operatorname{diag}(d_i^2)$ is diagonal. The problem (1.1) is then reduced to the form

$$Cy \equiv D^{-1}L^{-1}\Pi^T A \Pi L^{-T} D^{-1} y = \lambda y, \qquad y = DL^T \Pi^T x. \tag{1.3}$$

Any method for solving the symmetric eigenvalue problem can now be applied to C [6], [19]. In LAPACK's `xSYGV` driver, (1.1) is solved by applying the QR algorithm to (1.3). MATLAB 6's `eig` function does likewise when it is given a symmetric definite generalized eigenproblem. As is well known, when B is ill conditioned numerical stability can be lost in the Cholesky-based method. However, it is also known that methods based on factorizing B and converting to a standard eigenvalue problem have some attractive features. In reference to the method that uses a spectral decomposition of B, Wilkinson [25, p. 344] states that

> In the ill-conditioned case the method of §68 has certain advantages in that "all the condition of B" is concentrated in the small elements of D. The matrix P of (68.5) [our C in (1.3)] has a certain number of rows and columns with large elements (corresponding to small d_{ii}) and eigenvalues of $(A - \lambda B)$ of normal size are more likely to be preserved.

In this work we aim to give new insight into the numerical behavior of the Cholesky method.

First, we make a simple but important observation about numerical stability. Assume that the Cholesky factorization is computed exactly and set $\Pi = I$ without loss of generality. We compute $\widehat{C} = C + \Delta C_1$ where, at best, ΔC_1 satisfies a bound of the form

$$|\Delta C_1| \le c_n u |D^{-1}||L^{-1}||A||L^{-T}||D^{-1}|,$$

where c_n is a constant and u is the unit roundoff (see section 3 for the floating point arithmetic model). Here, $|A| = (|a_{ij}|)$. Solution of the eigenproblem for $\widehat{C}$ can be assumed to yield the exact eigensystem of $\widehat{C} + \Delta C_2$ for some ΔC_2. Therefore the computed eigensystem is the exact eigensystem of

$$C + \Delta C_1 + \Delta C_2 = D^{-1}L^{-1}\big(A + \Delta A\big)L^{-T}D^{-1}, \qquad \Delta A = LD(\Delta C_1 + \Delta C_2)DL^T,$$

and

$$\begin{aligned} |\Delta A| &\le |L||D|\big(c_n u |D^{-1}||L^{-1}||A||L^{-T}||D^{-1}| + |\Delta C_2|\big)|D||L^T| \\ &\le c_n u |L||L^{-1}||A||L^{-T}||L^T| + |L||D||\Delta C_2||D||L^T|. \end{aligned} \tag{1.4}$$

If we are using complete pivoting in the Cholesky factorization then $|l_{ij}| \le 1$ for $i > j$ and

$$d_1^2 \ge \cdots \ge d_n^2 > 0. \tag{1.5}$$

Hence [14, Theorem 8.13]

$$\kappa_p(L) = \|L\|_p\|L^{-1}\|_p \le n2^{n-1}, \qquad p = 1, 2, \infty \tag{1.6}$$

(with approximate equality achieved for L^T the Kahan matrix [14, p. 161]), and so the first term in (1.4) is bounded independently of $\kappa(B)$. The second term will have the same property provided that ΔC_2 satisfies a bound of the form

$$|\Delta C_2| \le |D^{-1}|f(|A|, |L^{-1}|, u)|D^{-1}|,$$

where f is a matrix depending on $|A|$, $|L^{-1}|$, and u, but not $|D^{-1}|$.

If nothing more is known about ΔC_2 than that $\|\Delta C_2\| \le c_n u\|C\|$ (corresponding to using a normwise backward stable eigensolver for C), then the best bound we can obtain in terms of the original data is of the form

$$\|\Delta A\| \le g(n)u\kappa(B)\|A\|. \tag{1.7}$$

However, this analysis shows that there is hope for obtaining a bound without the factor $\kappa(B)$ if the eigensolver for C respects the scaling of C when D is ill conditioned. The QL variant of the QR algorithm has this property in many instances, since when D is ill conditioned the inequalities (1.5) imply that C is graded upward (that is, its elements generally increase from top left to bottom right) and the backward error matrix for the QL algorithm[1] then tends to be graded in the same way [19, chapter 8], [21, p. 337]. However, this is a heuristic and we know of no precise results.

In this work we show that if, instead of the QL and QR algorithms, the Jacobi method is applied to C, then we can derive rigorous backward error bounds that can be significantly smaller than bounds involving a factor $\kappa(B)$ when B is ill conditioned. We also give experimental evidence of the benefits of pivoting in the Cholesky–QR method.

Wilkinson [26] expressed the view that for most of the standard problems in numerical linear algebra iterative refinement is a valuable tool for which it is worth developing software. We investigate iterative refinement as a means for improving the backward errors of eigenpairs computed by the Cholesky–QR and Cholesky–Jacobi methods.

The organization of the paper is as follows. In section 2 we describe the Cholesky–Jacobi method and in section 3 we give a detailed rounding error analysis, making use of a diagonal scaling idea of Anjos, Hammarling, and Paige [2]. In section 4 we show how fixed precision iterative refinement can be used to improve the stability of selected eigenpairs. Section 5 contains a variety of numerical examples. In particular, we describe a vibration problem from structural engineering where B is ill conditioned yet our backward error bounds for the Cholesky–Jacobi method are found to be of order u, and we give examples where ill condition of B does cause instability of the method but iterative refinement cures the instability. Conclusions are given in section 6.

In our analysis $\|\cdot\|$ denotes any vector norm and the corresponding subordinate matrix norm, while $\|\cdot\|_2$ and $\|\cdot\|_F$ denote the 2-norm and the Frobenius norm, respectively.

[1] For the original QR algorithm, we need C to be graded downward. However, the distinction is unimportant for our purposes since LAPACK's routines for the QR algorithm [1] include a strategy for switching between the QL and QR variants and thus automatically take advantage of either form of grading.

2. Method outline. The Cholesky–Jacobi method computes the Cholesky factorization with complete pivoting (1.2), forms

$$H_0 = D^{-1}L^{-1}\Pi^T A \Pi L^{-T} D^{-1} \tag{2.1}$$

in (1.3), and then applies Jacobi's method for the symmetric eigenproblem to H_0. Peters and Wilkinson [20] note that a variant of this method in which the Cholesky factorization of B is replaced by a spectral decomposition, computed also by the Jacobi method, was used by G. H. Golub on the Illiac at the University of Illinois in the 1950s.

Jacobi's method constructs a sequence of similar matrices starting with H_0. An orthogonal transformation is applied at each step,

$$H_{k+1} = Q_k^T H_k Q_k$$

in such a way that H_k tends to diagonal form $\Lambda = \operatorname{diag}(\lambda_i)$ as $k \to \infty$. Denoting by $Q = Q_0 Q_1 \ldots$ the product of the orthogonal transformations that diagonalizes H_0 and writing $X = \Pi L^{-T} D^{-1} Q$, we have, overall,

$$X^T A X = \Lambda, \quad X^T B X = I. \tag{2.2}$$

Thus X simultaneously diagonalizes A and B and is also easily seen to be a matrix of eigenvectors.

Now we describe the method in more detail. At the kth stage let Q_k be a Jacobi rotation in the (i, j) plane $(i \le j)$ such that $Q_k^T H_k Q_k$ has zeros in positions (i, j) and (j, i). Using MATLAB notation,

$$Q_k([i\ j], [i\ j]) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \tag{2.3}$$

where $c = \cos\theta$ and $s = \sin\theta$ are obtained from [12, section 8.4.2] (with $\operatorname{sign}(0) = 1$)

$$\tau = \frac{h_{jj} - h_{ii}}{2h_{ij}}, \tag{2.4}$$

$$t = \frac{\operatorname{sign}(\tau)}{|\tau| + \sqrt{1+\tau^2}}, \tag{2.5}$$

$$c = \frac{1}{\sqrt{1+t^2}}, \quad s = tc. \tag{2.6}$$

The corresponding rotation angle θ satisfies $|\theta| \le \pi/4$; choosing a small rotation angle is essential for the convergence theory [19, chapter 9]. We choose the index pairs (i, j) from a row cyclic ordering, in which a complete sweep has the form

$$(i, j) = (1,2), \ldots, (1,n), (2,3), \ldots, (2,n), \ldots, (n-1, n). \tag{2.7}$$

For this ordering and the choice of angle above, the Jacobi method converges quadratically [12, section 8.4.4], [19, section 9.4].

When forming $H_{k+1} = Q_k^T H_k Q_k = (\widetilde{h}_{ij})$ we explicitly set $\widetilde{h}_{ij} = 0$ and compute the new diagonal elements from [19, equation (9.9)]

$$\widetilde{h}_{ii} = h_{ii} - h_{ij}t, \tag{2.8}$$

$$\widetilde{h}_{jj} = h_{jj} + h_{ij}t, \tag{2.9}$$

where t is given in (2.5). The complete algorithm is summarized as follows.

ALGORITHM 2.1 (Cholesky–Jacobi method). Given $A, B \in \mathbb{R}^{n\times n}$ with A symmetric and B symmetric positive definite, this algorithm calculates the eigenvalues λ_i and corresponding eigenvectors x_i of the pair (A, B).

1. Compute the Cholesky factorization with complete pivoting $\Pi^T B \Pi = LD^2L^T$.
   Form $H = D^{-1}L^{-1}\Pi^T A \Pi L^{-T}D^{-1}$ by solving triangular systems.
   $X = \Pi L^{-T}D^{-1}$.
2. % Jacobi's method
   done_rot = true
   while done_rot = true
   done_rot = false
   for $i = 1\colon n$
   for $j = i+1\colon n$
   $(*)$ if $|h_{ij}| > u\sqrt{|h_{ii}h_{jj}|}$
   done_rot = true
   Form $Q_{ij} \equiv Q_k([i\ j],[i\ j])$ using (2.3)–(2.6).
   ind $= [1\colon i-1, i+1\colon j-1, j+1\colon n]$
   $H([i\ j], \text{ind}) = Q_{ij}^T H([i\ j], \text{ind})$
   $H(\text{ind}, [i\ j]) = H(\text{ind}, [i\ j])Q_{ij}$
   $H([i\ j],[i\ j]) = \begin{bmatrix} \widetilde{h}_{ii} & 0 \\ 0 & \widetilde{h}_{jj} \end{bmatrix}$ using (2.8), (2.9)
   $X(:, [i\ j]) = X(:, [i\ j])Q_{ij}$
   end
   end
   end
   end
   $\lambda_i = h_{ii}$, $x_i = X(:, i)$, $i = 1\colon n$

The test $(*)$ for whether to apply a rotation is adapted from the one used for Jacobi's method for a symmetric positive definite matrix [7]—we have added absolute values inside the square root since h_{ii} and h_{jj} can be negative. This test is too stringent in general and can cause the algorithm not to converge, but we have found it generally works well, and so we used it in our experiments in order to achieve the best possible numerical behavior.

3. Error analysis. Now we give an error analysis for Algorithm 2.1, with the aim of obtaining an error bound better than (1.7). We use the standard model for floating point arithmetic

$$fl(x \operatorname{op} y) = (x \operatorname{op} y)(1+\delta_1) = \frac{x \operatorname{op} y}{1+\delta_2}, \quad |\delta_1|, |\delta_2| \le u, \quad \operatorname{op} = +, -, *, /,$$
$$fl(\sqrt{x}) = \sqrt{x}(1+\delta), \quad |\delta| \le u,$$

where u is the unit roundoff. We will make use of the following lemma [14].

LEMMA 3.1. *If $|\delta_i| \le u$ and $\rho_i = \pm 1$ for $i = 1\colon n$, and $nu < 1$, then*

$$\prod_{i=1}^{n}(1+\delta_i)^{\rho_i} = 1+\theta_n, \quad \textit{where} \quad |\theta_n| \le \frac{nu}{1-nu} =: \gamma_n.$$

We define

$$\tilde{\gamma}_k = \frac{pku}{1-pku},$$

where p denotes a small integer constant whose exact value is unimportant. We will also write $\tilde{\theta}_k$ to denote a quantity satisfying $|\tilde{\theta}_k| \leq \tilde{\gamma}_k$. Computed quantities are denoted with a hat.

We consider first the second part of Algorithm 2.1, beginning with the construction of the Jacobi rotation.

LEMMA 3.2. *Let a Jacobi rotation Q_k be constructed using (2.4)–(2.6) so that $Q_k^T H_k Q_k$ has zeros in the (i,j) and (j,i) positions. The computed $\widehat{c}$, $\widehat{s}$, and $\widehat{t}$ satisfy*

$$\widehat{c} = c(1+\tilde{\theta}_1), \quad \widehat{s} = s(1+\tilde{\theta}_1'), \quad \widehat{t} = t(1+\tilde{\theta}_1''),$$

where c, s, and t are the exact values for H_k.

Proof. The proof is straightforward. ∎

In most of the rest of our analysis we will assume that the computed $\widehat{c}$, $\widehat{s}$, and $\widehat{t}$ are exact. It is easily checked that, in view of Lemma 3.2, this simplification does not affect the bounds.

LEMMA 3.3. *If one step of Jacobi's method is performed in the (i,j) plane on the matrix H_m then the computed $\widehat{H}_{m+1}$ satisfies*

$$\widehat{H}_{m+1} = Q_m^T \left(H_m + \Delta H_m\right) Q_m,$$

where the elements of ΔH_m are bounded componentwise by

$$\left.\begin{array}{l} |\Delta h_{ik}| \leq \tilde{\gamma}_1 \left(|h_{ik}| + 2|sc||h_{jk}|\right) \\ |\Delta h_{jk}| \leq \tilde{\gamma}_1 \left(|h_{jk}| + 2|sc||h_{ik}|\right) \end{array}\right\} \quad k \neq i,j,$$

and

$$\begin{aligned} |\Delta h_{ii}| &\leq \tilde{\gamma}_1 \left(c^2|h_{ii}| + |s/c||h_{ij}| + s^2|h_{jj}|\right), \\ |\Delta h_{ij}|, |\Delta h_{ji}| &\leq \tilde{\gamma}_1 \left(|sc||h_{ii}| + 2s^2|h_{ij}| + |sc||h_{jj}|\right), \\ |\Delta h_{jj}| &\leq \tilde{\gamma}_1 \left(s^2|h_{ii}| + |s/c||h_{ij}| + c^2|h_{jj}|\right). \end{aligned}$$

Proof. For the duration of the proof let $Q_m := Q_m([i\ j],[i\ j])$. Writing $H_m = (h_{ij})$ and $\widehat{H}_{m+1} = (\widehat{h}_{ij})$ and using a standard result for matrix–vector multiplication [14, section 3.5], we have, for $k \neq i,j$,

$$\begin{aligned} \begin{bmatrix} \widehat{h}_{ik} \\ \widehat{h}_{jk} \end{bmatrix} &= fl\left(Q_m^T \begin{bmatrix} h_{ik} \\ h_{jk} \end{bmatrix}\right), \\ &= \left(Q_m + \Delta Q_m\right)^T \begin{bmatrix} h_{ik} \\ h_{jk} \end{bmatrix}, \quad |\Delta Q_m| \leq \tilde{\gamma}_1 |Q_m|, \\ &=: Q_m^T \left(\begin{bmatrix} h_{ik} \\ h_{jk} \end{bmatrix} + \begin{bmatrix} \Delta h_{ik} \\ \Delta h_{jk} \end{bmatrix}\right). \end{aligned}$$

Then

$$\begin{aligned} \begin{bmatrix} |\Delta h_{ik}| \\ |\Delta h_{jk}| \end{bmatrix} &\leq |Q_m||\Delta Q_m^T| \begin{bmatrix} |h_{ik}| \\ |h_{jk}| \end{bmatrix} \\ &\leq \tilde{\gamma}_1 |Q_m||Q_m^T| \begin{bmatrix} |h_{ik}| \\ |h_{jk}| \end{bmatrix} \\ &= \tilde{\gamma}_1 \begin{bmatrix} 1 & 2|sc| \\ 2|sc| & 1 \end{bmatrix} \begin{bmatrix} |h_{ik}| \\ |h_{jk}| \end{bmatrix}, \end{aligned}$$

which gives the first two bounds. We calculate the elements at the intersection of rows and columns i and j using

$$\widehat{h}_{ii} = fl(h_{ii} - h_{ij}t) = (1+\tilde{\theta}_1)h_{ii} - (1+\tilde{\theta}_1)h_{ij}t,$$
$$\widehat{h}_{jj} = fl(h_{jj} + h_{ij}t) = (1+\tilde{\theta}_1)h_{jj} + (1+\tilde{\theta}_1)h_{ij}t,$$

and by setting $\widehat{h}_{ij}$ and $\widehat{h}_{ji}$ to zero. The backward perturbations Δh_{ii}, Δh_{ij}, and Δh_{jj} satisfy

$$Q_m^T \left(\begin{bmatrix} h_{ii} & h_{ij} \\ h_{ij} & h_{jj} \end{bmatrix} + \begin{bmatrix} \Delta h_{ii} & \Delta h_{ij} \\ \Delta h_{ij} & \Delta h_{jj} \end{bmatrix} \right) Q_m = \begin{bmatrix} \widehat{h}_{ii} & 0 \\ 0 & \widehat{h}_{jj} \end{bmatrix},$$

which can be expressed as

$$\begin{aligned} \begin{bmatrix} \Delta h_{ii} & \Delta h_{ij} \\ \Delta h_{ij} & \Delta h_{jj} \end{bmatrix} &= Q_m \begin{bmatrix} \widehat{h}_{ii} & 0 \\ 0 & \widehat{h}_{jj} \end{bmatrix} Q_m^T - \begin{bmatrix} h_{ii} & h_{ij} \\ h_{ij} & h_{jj} \end{bmatrix} \\ &= \begin{bmatrix} c^2\widehat{h}_{ii} + s^2\widehat{h}_{jj} & -sc\widehat{h}_{ii} + sc\widehat{h}_{jj} \\ -sc\widehat{h}_{ii} + sc\widehat{h}_{jj} & s^2\widehat{h}_{ii} + c^2\widehat{h}_{jj} \end{bmatrix} - \begin{bmatrix} h_{ii} & h_{ij} \\ h_{ij} & h_{jj} \end{bmatrix}. \end{aligned}$$

Substituting in for $\widehat{h}_{ii}$ and $\widehat{h}_{jj}$ and taking absolute values we obtain the second group of inequalities. (Note that $\Delta h_{ij} = \Delta h_{ji} = 0$ if c and s are exact, so by bounding Δh_{ij} and Δh_{ji} in this way we are allowing for inexact c and s.) ∎

In the next lemma we show that in the first rotation of Jacobi's method in Algorithm 2.1 a factor D^{-1} can be scaled out of the backward error, leaving a term that we can bound. We make use of the identity

$$sc = \frac{h_{ij}}{\sqrt{4h_{ij}^2 + (h_{ii} - h_{jj})^2}}, \tag{3.1}$$

which comes from manipulating the equations defining a Jacobi rotation and solving for $sc = \frac{1}{2}\sin 2\theta$ in terms of $\tan 2\theta$. In this result, $A_0 \equiv L^{-1}\Pi^T A \Pi L^{-T}$ in (2.1).

LEMMA 3.4. *Given a symmetric A_0 and a positive diagonal matrix $D_0 = \operatorname{diag}(d_i^2)$, suppose we perform one step of Jacobi's method in the (i,j) plane on $H_0 = D_0^{-1}A_0D_0^{-1}$, obtaining $H_1 = Q_0^T H_0 Q_0$. Then*

$$\widehat{H}_1 = fl(Q_0^T \widehat{H}_0 Q_0) = Q_0^T D_0^{-1}(A_0 + \Delta A_0)D_0^{-1}Q_0, \tag{3.2}$$

where

$$\|\Delta A_0\|_2 \le \tilde{\gamma}_n(1+2\omega_0)\|A_0\|_2, \tag{3.3}$$

with

$$\omega_0 = |sc|\max(\rho, 1/\rho), \qquad \rho = d_i/d_j.$$

Proof. We start by forming the matrix $H_0 = (h_{ij})$. Since we are given the squared diagonal elements d_i^2 we have

$$\begin{aligned} \widehat{h}_{ij} &= fl\left(a_{ij}/\sqrt{d_i^2 d_j^2}\right) \\ &= (1+\theta_3)a_{ij}/(d_i d_j) = (1+\theta_3)h_{ij} \\ &=: \widehat{a}_{ij}/(d_i d_j). \end{aligned}$$

Thus these initial errors can be thrown onto A_0: $\widehat{H}_0 = D_0^{-1}(A_0 + \Delta_1)D_0^{-1}$, where $|\Delta_1| \le \gamma_3|A_0|$. The errors in applying one step of Jacobi's method to $\widehat{H}_0$ can be expressed as a backward perturbation ΔH_0 to $\widehat{H}_0$ using Lemma 3.3. The corresponding perturbation of $\widehat{A}_0 = A_0 + \Delta_1$ is $\Delta_2 = D_0 \Delta H_0 D_0$, so we simply scale the componentwise perturbation bounds of Lemma 3.3. We find

$$\left.\begin{array}{l} |(\Delta_2)_{ik}| \le \tilde{\gamma}_1 \left(|\widehat{a}_{ik}| + 2|sc||\widehat{a}_{jk}|\rho\right) \\ |(\Delta_2)_{jk}| \le \tilde{\gamma}_1 \left(|\widehat{a}_{jk}| + 2|sc||\widehat{a}_{ik}|/\rho\right) \end{array}\right\} \quad k \ne i, j,$$

$$|(\Delta_2)_{ii}| \le \tilde{\gamma}_1 \left(c^2|\widehat{a}_{ii}| + |s/c||\widehat{a}_{ij}|\rho + s^2|\widehat{a}_{jj}|\rho^2\right), \tag{3.4}$$

$$|(\Delta_2)_{ij,ji}| \le \tilde{\gamma}_1 \left(|sc||\widehat{a}_{ii}|/\rho + 2s^2|\widehat{a}_{ij}| + |sc||\widehat{a}_{jj}|\rho\right),$$

$$|(\Delta_2)_{jj}| \le \tilde{\gamma}_1 \left(s^2|\widehat{a}_{ii}|/\rho^2 + |s/c||\widehat{a}_{ij}|/\rho + c^2|\widehat{a}_{jj}|\right). \tag{3.5}$$

We now work to remove the potentially large ρ^2 and $1/\rho^2$ terms. We can rewrite (3.1) as

$$sc = \frac{\frac{a_{ij}}{d_i d_j}}{\sqrt{4\frac{a_{ij}^2}{d_i^2 d_j^2} + \left(\frac{a_{ii}}{d_i^2} - \frac{a_{jj}}{d_j^2}\right)^2}} = \frac{\rho a_{ij}}{\sqrt{\left(a_{ii} - \rho^2 a_{jj}\right)^2 + 4\rho^2 a_{ij}^2}}. \tag{3.6}$$

Further manipulation yields

$$|a_{jj}|\rho^2 \le |a_{ii}| + \sqrt{\frac{a_{ij}^2\rho^2}{(sc)^2} - 4a_{ij}^2\rho^2} = |a_{ii}| + \rho|a_{ij}|\sqrt{\frac{1}{(sc)^2} - 4}.$$

Therefore

$$s^2|a_{jj}|\rho^2 \le s^2|a_{ii}| + \rho|a_{ij}|\sqrt{t^2 - 4s^4}. \tag{3.7}$$

A similar manipulation of (3.1) (or a symmetry argument) gives

$$s^2|a_{ii}|/\rho^2 \le s^2|a_{jj}| + \frac{|a_{ij}|}{\rho}\sqrt{t^2 - 4s^4}. \tag{3.8}$$

Since $\widehat{a}_{ij} = a_{ij}(1 + \theta_3)$ there is no harm in replacing a_{ij} by $\widehat{a}_{ij}$ in (3.7) and (3.8). Since $\theta \in [-\pi/4, \pi/4]$ we have

$$\sqrt{t^2 - 4s^4} + |s/c| = 2|sc|, \tag{3.9}$$

and hence (3.4) and (3.5) may be bounded by

$$\left.\begin{array}{l} |(\Delta_2)_{ii}| \le \tilde{\gamma}_1 \left(|\widehat{a}_{ii}| + 2|sc||\widehat{a}_{ij}|\rho\right), \\ |(\Delta_2)_{jj}| \le \tilde{\gamma}_1 \left(|\widehat{a}_{jj}| + 2|sc||\widehat{a}_{ij}|/\rho\right). \end{array}\right.$$

Setting $\Delta A = \Delta_1 + \Delta_2$ and using these componentwise bounds we obtain the overall bound given in (3.3). □

Lemma 3.4 shows that the Jacobi rotation results in a small backward perturbation to A_0 provided that ω_0 is of order 1. We see from (3.6) that in normal circumstances sc is proportional to $\min(\rho, 1/\rho)$, which keeps ω_0 small. However, in special situations ω_0 can be large, for example, when $|a_{ii} - \rho^2 a_{jj}| \ll \rho|a_{ij}|$ with ρ large, which requires that $|a_{jj}|$ be much smaller than $|a_{ij}|$ and B be ill conditioned.

By combining Lemma 3.4 with subsequent applications of Lemma 3.3 we find that after m steps of Jacobi's method on $H_0 = D_0^{-1}A_0D_0^{-1}$ we have

$$\widehat{H}_m = Q_{m-1}^T \dots Q_0^T (H_0 + \Delta_0) Q_0 \dots Q_{m-1},$$

where

$$\begin{aligned}\Delta_0 &= D_0^{-1}\Delta A_0 D_0^{-1} + \sum_{k=1}^{m-1} Q_0 \dots Q_{k-1}\Delta H_k Q_{k-1}^T \dots Q_0^T \\ &= D_0^{-1}\left(\Delta A_0 + \sum_{k=1}^{m-1} D_0 Q_0 \dots Q_{k-1}\Delta H_k Q_{k-1}^T \dots Q_0^T D_0\right) D_0^{-1}.\end{aligned}$$

The ΔH_k are bounded as in Lemma 3.3. We would like to bound the term in parentheses by a multiple of $u\|A_0\|_2$, but simply taking norms leads to an unsatisfactory $\kappa(D_0^2)$ factor. To obtain a better bound we introduce, purely for theoretical purposes, a scaling to $\widehat{H}_k$ at each stage of the iteration. For an arbitrary nonsingular diagonal D_k we write

$$\begin{aligned}\|D_0Q_0 \dots Q_{k-1}\Delta H_k Q_{k-1}^T \dots Q_0^T D_0\|_2 &= \|D_0Q_0 \dots Q_{k-1}D_k^{-1} \cdot D_k\Delta H_k D_k \\ &\qquad \cdot D_k^{-1}Q_{k-1}^T \dots Q_0^T D_0\|_2 \\ &\le \min_{D_k \text{ diag}} \left(\|D_0Q_0 \dots Q_{k-1}D_k^{-1}\|_2^2 \|D_k\Delta H_k D_k\|_2\right) \\ &= \min_{D_k \text{ diag}} \left(\|N_k^{-T}\|_2^2 \|D_k\Delta H_k D_k\|_2\right),\end{aligned}$$

where

$$N_k = D_0^{-1}Q_0 \dots Q_{k-1}D_k. \tag{3.10}$$

Define

$$A_k := N_k^T A_0 N_k = D_k H_k D_k. \tag{3.11}$$

By applying Lemma 3.4 to a rotation on H_k, we can see that

$$\|D_k\Delta H_k D_k\|_2 \le \tilde{\gamma}_n(1 + 2\omega_k)\|A_k\|_2, \tag{3.12}$$

where

$$\omega_k = |s_k c_k| \max(\rho_k, 1/\rho_k), \qquad \rho_k = d_i^{(k)}/d_j^{(k)},$$

with a subscript k denoting quantities on the kth step and where $D_k = \operatorname{diag}(d_i^{(k)})$. One way to proceed is to choose D_k to minimize $\kappa_2(M_{k-1})$, where

$$M_{k-1} = D_{k-1}^{-1}Q_{k-1}D_k. \tag{3.13}$$

Notice that

$$N_k = M_0 \dots M_{k-1}. \tag{3.14}$$

This idea is based on an algorithm of Anjos, Hammarling, and Paige [2] that avoids explicitly inverting any of the D_k and uses transformation matrices of the form in (3.13)

to diagonalize A while retaining the diagonal form of D_0. The algorithm computes the congruence transformations

$$A_{k+1} = M_k^T A_k M_k, \quad D_{k+1}^2 = M_k^T D_k^2 M_k,$$

where D_k is diagonal for all k and A_k tends to diagonal form as $k \to \infty$. The difference between our approach and that in [2] is that we form $H_0 = D_0^{-1} A_0 D_0^{-1}$ and use D_k in the analysis to obtain stronger error bounds, whereas in [2], in an effort to apply only well-conditioned similarity transformations, H_0 is never formed but M_k is computed and applied in the algorithm (and no error analysis is given in [2]).

Now we discuss the choice of D_k, drawing on analysis from [2]. Since Q_{k-1} is a rotation in the (i, j) plane, we choose D_k to be identical to D_{k-1} in all but the ith and jth diagonal entries. Thus M_{k-1} is the identity matrix except in the (i, j) plane, in which

$$M_{ij} = M([i\ j], [i\ j]) = \begin{bmatrix} d_i^{-1} & 0 \\ 0 & d_j^{-1} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \widetilde{d}_i & 0 \\ 0 & \widetilde{d}_j \end{bmatrix},$$

where we are writing

$$D_{k-1} = \operatorname{diag}(d_i), \quad D_k = \operatorname{diag}(\widetilde{d}_i).$$

We now choose D_k to minimize the 2-norm condition number $\kappa_2(M_{ij})$. It can be shown that for any 2×2 matrix, G, say,

$$\kappa_2(G) = \sigma_1(G)/\sigma_2(G) = \left(\phi^2 + \sqrt{\phi^4 - 4\delta^2}\right)/2\delta,$$

where $\phi = \|G\|_F$, $\delta = |\det(G)|$ and $\sigma_1(G) \geq \sigma_2(G)$ are the singular values of G. Using $\kappa_F(G) = \phi^2/\delta$, we obtain

$$\kappa_2(G) = \left(\kappa_F(G) + \sqrt{\kappa_F(G)^2 - 4}\right)/2,$$

so clearly $\kappa_2(G)$ has its minimum when $\kappa_F(G)$ does. Therefore it is only necessary to analyze $\kappa_F(M_{ij})$ in order to find the minimum of $\kappa_2(M_{ij})$. For M_{ij} we have

$$\begin{aligned} \phi^2 &= s^2 \left((\widetilde{d}_i/d_j)^2 + (\widetilde{d}_j/d_i)^2\right) + c^2 \left((\widetilde{d}_j/d_j)^2 + (\widetilde{d}_i/d_i)^2\right), \\ \delta &= \det(D_{k-1}^{-1}) \det(D_k) = (\widetilde{d}_i \widetilde{d}_j)/(d_i d_j). \end{aligned}$$

Setting $\xi = \widetilde{d}_i/\widetilde{d}_j$ we have

$$\kappa_F(M_{ij}) = \phi^2/\delta = \left(c^2(\rho^2 + \xi^2) + s^2(\rho^2\xi^2 + 1)\right)/(\rho\xi).$$

This is an equation with only one unknown, ξ. The minimum of $\kappa_F(M_{ij})$ over ξ occurs at

$$\xi_{\text{opt}}^2 = \left(s^2 + \rho^2 c^2\right) / \left(c^2 + \rho^2 s^2\right),$$

which gives the values

$$\begin{aligned} \kappa_F(M_{ij})_{\min} &= 2\sqrt{1 + s^2 c^2 \left(\rho - \rho^{-1}\right)^2}, \\ \kappa_2(M_{ij})_{\min} &= |sc(\rho - \rho^{-1})| + \sqrt{1 + s^2 c^2 \left(\rho - \rho^{-1}\right)^2}. \end{aligned} \tag{3.15}$$

Knowing the ratio $\widetilde{d}_i/\widetilde{d}_j$ that minimizes $\kappa_2(M_0)$, we now have to choose $\widetilde{d}_j$ and then set $\widetilde{d}_i = \widetilde{d}_j \xi_{\mathrm{opt}}$. We set $\|D_k\|_F = \|D_{k-1}\|_F$, or more simply,

$$d_i^2 + d_j^2 = \widetilde{d}_i^2 + \widetilde{d}_j^2 = \left(\xi_{\mathrm{opt}}^2 + 1\right) \widetilde{d}_j^2. \tag{3.16}$$

This yields the values

$$\begin{aligned} \widetilde{d}_i^2 &= c^2 d_i^2 + s^2 d_j^2, \\ \widetilde{d}_j^2 &= c^2 d_j^2 + s^2 d_i^2 \end{aligned} \tag{3.17}$$

and the matrix

$$M_{ij} = \begin{bmatrix} c\sqrt{c^2 + s^2/\rho^2} & s\sqrt{s^2 + c^2/\rho^2} \\ -s\sqrt{s^2 + c^2\rho^2} & c\sqrt{c^2 + s^2\rho^2} \end{bmatrix}. \tag{3.18}$$

Clearly,

$$\min(d_i^2, d_j^2) \le \widetilde{d}_k^2 \le \max(d_i^2, d_j^2), \qquad k = i, j. \tag{3.19}$$

We note for later reference that a direct calculation reveals

$$\|M_{ij}^{-1}\|_F = \sqrt{2}. \tag{3.20}$$

It is also interesting to note that M_{ij} has columns of equal 2-norm. This is not surprising in view of a result of van der Sluis [24], which states that scaling the columns of an $n \times n$ matrix to have equal 2-norms produces a matrix with 2-norm condition number within a factor $\sqrt{n}$ of the minimum over all column scalings.

To complete our analysis we need to bound $\|A_k\|_2$ and $\|N_i^{-1}\|_2$.

3.1. Growth of A_m. We now bound $\|A_m\|_2$, which appears in the bound (3.12). We consider the growth over one step from $A_m = (a_{ij})$ to $A_{m+1} = (\widetilde{a}_{ij}) = M_m^T A_m M_m$, as measured by $\phi_m = \max_{i,j} |\widetilde{a}_{ij}| / \max_{i,j} |a_{ij}|$. By rewriting (2.8) and (2.9) in terms of A_k, and using (3.11) and (3.17), we can show that

$$|\widetilde{a}_{ii}| \le c^2|a_{ii}| + s^2|a_{ii}|/\rho^2 + |a_{ij}| \left(\frac{|s^3|}{c\rho} + |sc|\rho \right), \tag{3.21}$$

$$|\widetilde{a}_{jj}| \le c^2|a_{jj}| + s^2|a_{jj}|\rho^2 + |a_{ij}| \left(\frac{|sc|}{\rho} + \frac{|s^3|}{c}\rho \right). \tag{3.22}$$

We would like to bound these two elements linearly in terms of $\max(\rho, 1/\rho)$ (recall that ρ can be greater than or less than 1). The troublesome terms in the bounds are $s^2|a_{jj}|\rho^2$ and $s^2|a_{ii}|/\rho^2$. Upon substitution of (3.7) and (3.8) in (3.21) and (3.22) we obtain bounds linear in ρ and $1/\rho$:

$$\begin{aligned} |\widetilde{a}_{ii}| &\le c^2|a_{ii}| + s^2|a_{jj}| + |a_{ij}| \left(\left(\sqrt{t^2 - 4s^4} + \frac{|s^3|}{c} \right) \frac{1}{\rho} + |sc|\rho \right), \\ |\widetilde{a}_{jj}| &\le c^2|a_{jj}| + s^2|a_{ii}| + |a_{ij}| \left(\left(\sqrt{t^2 - 4s^4} + \frac{|s^3|}{c} \right) \rho + \frac{|sc|}{\rho} \right). \end{aligned}$$

Using (3.9) we find that $\sqrt{t^2 - 4s^4} + |s^3|/|c| = |sc|$, and so

$$|\widetilde{a}_{ii}| \le c^2|a_{ii}| + s^2|a_{jj}| + |a_{ij}||sc| \left(\rho + 1/\rho\right), \tag{3.23}$$

$$|\widetilde{a}_{jj}| \le c^2|a_{jj}| + s^2|a_{ii}| + |a_{ij}||sc| \left(\rho + 1/\rho\right). \tag{3.24}$$

For the other affected elements in rows and columns i and j we have, for $k \neq i, j$,

$$\widetilde{a}_{ik} = \widetilde{a}_{ki} = a_{ik}c\sqrt{c^2 + s^2/\rho^2} - a_{jk}s\sqrt{s^2 + c^2\rho^2},$$
$$\widetilde{a}_{jk} = \widetilde{a}_{kj} = a_{ik}s\sqrt{s^2 + c^2/\rho^2} + a_{jk}c\sqrt{c^2 + s^2\rho^2}.$$

These elements can be bounded by

$$|\widetilde{a}_{ik}| \leq |a_{ik}|\left(c^2 + |sc|/\rho\right) + |a_{jk}|\left(s^2 + |sc|\rho\right), \tag{3.25}$$
$$|\widetilde{a}_{jk}| \leq |a_{ik}|\left(s^2 + |sc|/\rho\right) + |a_{jk}|\left(c^2 + |sc|\rho\right). \tag{3.26}$$

The bounds (3.23)–(3.26) can all be written in the form

$$|\widetilde{a}_{pq}| \leq \max_{r,s} |a_{rs}|\big(1 + |sc|(\rho + 1/\rho)\big),$$

and so the growth of A_m over one step is bounded by

$$\phi_m \leq 1 + |sc|(\rho + 1/\rho) \leq 1 + 2|sc| \max(\rho, 1/\rho) = 1 + 2\omega_m.$$

The overall growth bound is

$$\pi_m := \frac{\|A_m\|_2}{\|A_0\|_2} \leq \sqrt{n} \prod_{i=0}^{m-1} \phi_i. \tag{3.27}$$

3.2. Bounding $\|N_i^{-1}\|_2$. Our final task is to bound

$$\mu_i := \|N_i^{-1}\|_2 = \|D_i^{-1} Q_{i-1}^T \dots Q_0^T D_0\|_2$$

(see (3.10)). We describe two different bounds. In view of (3.19),

$$\|D_{i+1}^{-1}\|_2 \leq \|D_i^{-1}\|_2 \leq \cdots \leq \|D_0^{-1}\|_2.$$

Thus, since $D_0 = D$, where B has the Cholesky factorization (1.2),

$$\mu_i^2 \leq \kappa_2(D)^2 \leq \kappa_2(L)\kappa_2(B).$$

However, the point of our analysis is to avoid a $\kappa_2(B)$ term in the bounds. As an alternative way of bounding μ_i we note that, from (3.14),

$$N_i^{-1} = M_{i-1}^{-1} \dots M_0^{-1}.$$

For the row cyclic ordering in (2.7) the congruence transformations can be reordered into $2n-3$ groups of up to $\lceil n/2 \rceil$ disjoint transformations $M_{j+1}, \dots, M_{j+p}$ such that, using (3.20),

$$\|M_{j+p}^{-1} \dots M_{j+1}^{-1}\|_2 \leq \sqrt{2}.$$

For example, a sweep of a 6×6 matrix can be divided into 9 groups of disjoint rotations:

$$\begin{bmatrix} - & 1 & 2 & 3 & 4 & 5 \\ & - & 3 & 4 & 5 & 6 \\ & & - & 5 & 6 & 7 \\ & & & - & 7 & 8 \\ & & & & - & 9 \\ & & & & & - \end{bmatrix}.$$

Here, an integer k in position (i,j) denotes that the (i,j) element is eliminated on the kth step by a rotation in the (i,j) plane, and all rotations on the kth step are disjoint. Hence we can bound μ_i by

$$\mu_i \le (\sqrt{2})^{2n-3} = 2^{n-3/2}.$$

Although exponential in n, this bound is independent of $\kappa_2(B)$.

3.3. Summary. Our backward error analysis shows that, upon convergence after m Jacobi rotations, Algorithm 2.1 has computed a diagonal Λ such that

$$X^T(A+\Delta A)X = \Lambda, \quad X^T(B+\Delta B)X = I \tag{3.28}$$

for some nonsingular X, where

$$\|\Delta A\|_2 \le \tilde{\gamma}_{n^2}\|A\|_2\left(\kappa_2(L)^2 + \sum_{k=0}^{m-1}\mu_k^2\big(1+2\omega_k\big)\pi_k\right), \tag{3.29a}$$

$$\|\Delta B\|_2 \le \tilde{\gamma}_{n^2}\|B\|_2. \tag{3.29b}$$

The term involving $\kappa_2(L)$ takes account of errors in the first stage of Algorithm 2.1 and follows from standard error analysis [14, chapter 10] of Cholesky factorization and the solution of triangular systems. Because of the complete pivoting, $\kappa(L)$ is bounded as in (1.6), and in practice it is usually small. Even when $\kappa(L)$ is large, its full effect tends not to be felt on the backward error, since triangular systems are typically solved to higher accuracy than the bounds suggest [14, chapter 8].

We do not have a bound better than exponential in n for the term μ_i^2, but this term has been less than 10 in virtually all our numerical tests. We showed in section 3.1 that the growth factor $\pi_k = \|A_k\|_2/\|A_0\|_2$ in (3.27) is certainly bounded by $\pi_k \le \sqrt{n}\prod_{i=0}^{k-1}(1+2\omega_i)$. The term

$$\omega_k = |s_k c_k|\max(\rho_k, 1/\rho_k) \le |s_k c_k|\kappa_2(D) \le |s_k c_k|\kappa_2(L)\kappa_2(B)^{1/2} \tag{3.30}$$

is the most important quantity in our analysis. A large value of ω_k, for some k, is the main indicator of instability in Algorithm 2.1.

We stress that our error bounds do not depend on the ordering (1.5), as should be expected since the Jacobi method is insensitive to the ordering of the diagonal of D. The purpose of pivoting in the Cholesky factorization is to keep L well conditioned and thereby concentrate any ill conditioning of B into D.

The conclusion from the error analysis is that Algorithm 2.1 has much better stability properties than the bound (1.7) suggests. When $\kappa_2(B)$ is large it is usually the case that small values of $|s_k c_k|$ cancel any large values of $\max(\rho_k, 1/\rho_k)$ (see the discussion following Lemma 3.4) and that π_k is also small, with a resulting small backward error bound.

For the particular version of the Cholesky–QR method in which the initial tridiagonalization of the QR algorithm is performed using Givens rotations, Davies [5] uses suitable modifications of the analysis presented here to derive analogues of (3.28) and (3.29) in which the terms $1+2\omega_k$ and π_k in (3.29) are squared (the definitions of w_k and π_k are unchanged, but of course the underlying rotations are different). Unfortunately, Householder transformations rather than Givens rotations are almost always used for the tridiagonalization and our error analysis is specific to rotations; therefore (1.7) remains the best error bound for the practically used Cholesky–QR method.

4. Iterative refinement. The relative normwise backward error of an approximate eigenpair $(\widetilde{x}, \widetilde{\lambda})$ of (1.1) is defined by

$$\eta(\widetilde{x}, \widetilde{\lambda}) = \min\{\epsilon : (A + \Delta A)\widetilde{x} = \widetilde{\lambda}(B + \Delta B)\widetilde{x}, \quad \|\Delta A\| \le \epsilon\|A\|, \quad \|\Delta B\| \le \epsilon\|B\|\}. \tag{4.1}$$

To evaluate the backward error we can use the explicit expression [11], [13]

$$\eta(\widetilde{x}, \widetilde{\lambda}) = \frac{\|r\|}{(|\widetilde{\lambda}|\,\|B\| + \|A\|)\|\widetilde{x}\|}, \tag{4.2}$$

where $r = \widetilde{\lambda}B\widetilde{x} - A\widetilde{x}$ is the residual. For symmetric A and B, we denote by $\eta^S(\widetilde{x}, \widetilde{\lambda})$ the backward error (4.1) with the additional constraint that the perturbations ΔA and ΔB are symmetric. Clearly $\eta^S(\widetilde{x}, \widetilde{\lambda}) \ge \eta(\widetilde{x}, \widetilde{\lambda})$. However, Higham and Higham [13] show that when $\widetilde{\lambda}$ is real, $\eta^S(\widetilde{x}, \widetilde{\lambda}) = \eta(\widetilde{x}, \widetilde{\lambda})$ for the 2-norm. Hence, for the symmetric definite generalized eigenproblem it is appropriate to use the general definition (4.1) and the formula (4.2).

The idea of using iterative refinement to improve numerical stability has been investigated for linear systems by several authors; see [14, chapter 11] for a survey and [15] for the most recent results. Iterative refinement has previously been used with residuals computed in extended precision to improve the accuracy of approximate solutions to the standard eigenproblem [8], [9], [22]. Tisseur [23] shows how iterative refinement can be used in fixed or extended precision to improve the forward and backward errors of approximate solutions to the generalized eigenvalue problem (GEP). She writes the GEP as

$$Ax = \lambda Bx, \qquad e_s^T x = 1 \text{ (for some fixed } s)$$

and applies Newton's method to the equivalent nonlinear equation problem

$$F\left(\begin{bmatrix} x \\ \lambda \end{bmatrix}\right) = \begin{bmatrix} (A - \lambda B)x \\ e_s^T x - 1 \end{bmatrix} : \mathbb{R}^{n+1} \to \mathbb{R}^{n+1}.$$

This requires solving linear systems whose coefficient matrices are the Jacobian

$$J\left(\begin{bmatrix} x \\ \lambda \end{bmatrix}\right) = \begin{bmatrix} A - \lambda B & -Bx \\ e_s^T & 0 \end{bmatrix}.$$

We use this technique with residuals computed in *fixed precision* to improve the backward errors of eigenpairs computed by Algorithm 2.1. We very briefly summarize the convergence results and two implementations of iterative refinement; full details may be found in [23].

If J is not too ill conditioned, the linear system solver is not too unstable, and the starting vector is sufficiently close to an eigenpair (x_*, λ_*), then iterative refinement by Newton's method in floating point arithmetic with residuals computed in fixed precision yields a refined eigenpair $(\widehat{x}, \widehat{\lambda})$ with backward error in the ∞-norm bounded by [23, Corollary 3.5]

$$\eta_\infty(\widehat{x}, \widehat{\lambda}) \le \tilde{\gamma}_n + u(3 + |\lambda|)\max\left(\frac{\|A\|_\infty}{\|B\|_\infty}, \frac{\|B\|_\infty}{\|A\|_\infty}\right). \tag{4.3}$$

This backward error bound is small if λ is of order 1 and the problem is well balanced, that is, $\|A\|_\infty \approx \|B\|_\infty$. If the problem is not well balanced, we can change the GEP

to make it so. We can scale the GEP to $(\alpha A)x = (\alpha\lambda)Bx$, where $\alpha = \|B\|_\infty/\|A\|_\infty$ and the backward error now depends on the size of $\overline{\lambda} = \alpha\lambda$. If $|\overline{\lambda}| \leq 1$, a small backward error is ensured, while for $|\overline{\lambda}| \geq 1$ we can consider the problem $Bx = \overline{\mu}\overline{A}x$, for which $|\overline{\mu}| \leq 1$. Practical experience shows that it is not necessary to scale or to reverse the problem—a backward error of order u is obtained as long as the starting vector is good enough for Newton's method to converge.

The following algorithm can be derived after some manipulation of the Newton equations [23].

ALGORITHM 4.1. Given A, B and an approximate eigenpair (x, λ) with $\|x\|_\infty = x_s = 1$, this algorithm applies iterative refinement to λ and x:

repeat until convergence
  $r = \lambda Bx - Ax$
  Form M: the matrix $A - \lambda B$ with column s replaced by $-Bx$
  Factor $PM = LU$ (LU factorization with partial pivoting)
  Solve $M\delta = r$ using the LU factors
  $\lambda = \lambda + \delta_s$; $\delta_s = 0$
  $x = x + \delta$
end

This algorithm is expensive as each iteration requires $O(n^3)$ flops for the factorization of M. By taking advantage of the eigendecomposition computed by Algorithm 2.1, the cost per iteration can be reduced to $O(n^2)$ flops [23].

ALGORITHM 4.2. Given A, B, X, and Λ such that $X^TAX = \Lambda$ and $X^TBX = I$, and an approximate eigenpair (x, λ) with $\|x\|_\infty = x_s = 1$, this algorithm applies iterative refinement to λ and x at a cost of $O(n^2)$ flops per iteration.

repeat until convergence
  $r = \lambda Bx - Ax$
  $D_\lambda = \Lambda - \lambda I$
  $d = -Bx - c_{\lambda s}$, where $c_{\lambda s}$ is the sth column of $A - \lambda B$
  $v = X^Td$; $f = X^Te_s$
  Compute Givens rotations J_k in the $(k, k+1)$ plane, such that
    $Q_1^Tv := J_1^T \ldots J_{n-1}^Tv = \|v\|_2 e_1$
  Compute orthogonal Q_2 such that
    $T = Q_2^TQ_1^T(D_\lambda + vf^T)$ is upper triangular
  $z = Q_2^TQ_1^TX^Tr$
  Solve $Tw = z$ for w
  $\delta = Xw$
  $\lambda = \lambda + \delta_s$; $\delta_s = 0$
  $x = x + \delta$
end

The computed $\widehat{X}$ from Algorithm 2.1 does not necessarily give a backward stable diagonalization of A and B. However, Tisseur [23] shows that instability in the solver does not affect the overall limiting accuracy and limiting backward error (4.3) when iterative refinement converges, although of course it may inhibit convergence. The price to be paid for the greater efficiency of Algorithm 4.2 over Algorithm 4.1 is less frequent and less rapid convergence.

5. Numerical results. In this section we give several examples to illustrate the behavior of Algorithm 2.1 and the sharpness of our backward error bounds, to show how the algorithm compares with the Cholesky–QR method, to show the need for pivoting in the Cholesky–QR method, and to show the benefits of iterative refinement.

TABLE 5.1
Terms from error analysis and backward error for Example 1.

ϵ	$\kappa_2(B)$	$\max \omega_k$	$\max \mu_k^2$	$\max \pi_k$	$\max \eta_2(\widehat{x}, \widehat{\lambda})$
10^{-1}	10^{7}	7.98e-1	3.33e0	3.12e0	1.31e-16
10^{-2}	10^{14}	1.90e0	4.38e0	7.02e0	5.35e-17
10^{-3}	10^{21}	2.38e0	4.67e0	1.04e1	3.50e-17

All our experiments were carried out in MATLAB 6, in which matrix computations are based on LAPACK; the unit roundoff is $u = 2^{-53} \approx 1.1 \times 10^{-16}$. (Our implementation of the Cholesky–QR method uses the MATLAB/LAPACK implementation of the QR algorithm and so employs Householder tridiagonalization.) In Algorithms 4.1 and 4.2 convergence was declared when $\eta_\infty(\widehat{x}, \widehat{\lambda}) \leq u$.

Example 1. Our first example illustrates how our backward error bounds can correctly predict perfect backward stability of Algorithm 2.1 despite large values of $\kappa_2(B)$. We take $A = H - I \in \mathbb{R}^{n \times n}$, where H is the Hilbert matrix, and $B = \mathrm{diag}(1, \epsilon, \epsilon^2, \ldots, \epsilon^{n-1})$. For $n = 8$ and $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}$, Table 5.1 shows the values of the terms appearing in the error analysis along with the maximum backward error over all the computed eigenpairs. The Cholesky–QR method is also stable on this example.

In a variation of this example we took $A = H$ and $B = \mathrm{diag}(\epsilon^{n-1}, \ldots, \epsilon, 1)$, with $n = 8$ and $\epsilon = 10^{-2}$. The computed eigenvalues from the Cholesky–Jacobi method and the Cholesky–QR method with pivoting both range from 10^{-9} to 10^{14} and the maximum backward error over all the computed eigenpairs is of order u. However, the Cholesky–QR method without pivoting produces two negative eigenvalues of order 10^{-2}, even though the exact eigenvalues are clearly positive, and the maximum backward error is of order 10^{-3}.

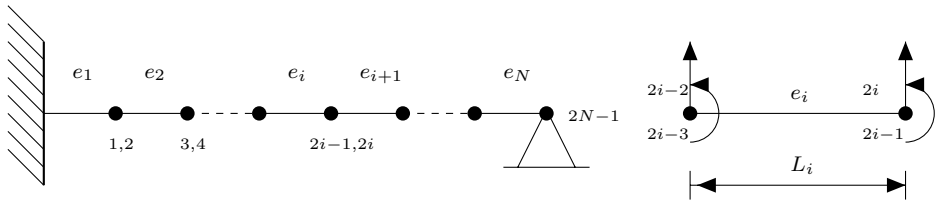
Example 2. This example is a structural engineering problem that again illustrates independence of our backward error bounds on $\kappa_2(B)$. We consider a cantilever beam as shown in Figure 5.1(a). We assume that the cantilever is rigid in its axial direction and that all the deformations are small. The boundary conditions are full-fixity at the base and zero translational displacement at the cantilever end. We also assume that the material properties and cross sections vary along the length of the beam. The equation of motion for the natural vibrations has the form

$$M\ddot{v} + Kv = 0,$$

where M denotes the symmetric positive definite mass inertia matrix and K the symmetric positive definite stiffness matrix. The finite element method leads to the generalized eigenvalue problem

$$K\phi = \lambda M\phi. \tag{5.1}$$

The cantilever is modeled with N finite elements. Each element has 4 degrees of freedom, namely, the two beam-end lateral displacements and the two beam-end rotations as shown in Figure 5.1(b). The length of the ith finite element e_i is taken to be L_i and its flexural characteristic to be $(EI)_i$, where E is the modulus of elasticity and I the moment of inertia. The global degrees of freedom are numbered as shown in Figure 5.1(a). If cubic Hermite interpolation polynomials are used to describe



(a) Geometry of supported cantilever beam. (b) Beam finite element.

FIG. 5.1. *Single span cantilever beam with supported end point.*

TABLE 5.2
Result for two instances of the cantilever beam problem.

$\kappa_2(M) = 3.9 \times 10^{10}$, $\kappa_2(L) = 1.8$

	$\max \omega_k$	$\max \mu_k^2$	$\max \pi_k$	$\max \eta_2(\widehat{x}, \widehat{\lambda})$
Cholesky–Jacobi	4.58e0	8.3e0	1.63e0	5.18e-17
Cholesky–QR (no pivoting)				5.10e-17
Cholesky–QR (with pivoting)				7.48e-17

$\kappa_2(M) = 6.7 \times 10^{6}$, $\kappa_2(L) = 2.2$

	$\max \omega_k$	$\max \mu_k^2$	$\max \pi_k$	$\max \eta_2(\widehat{x}, \widehat{\lambda})$
Cholesky–Jacobi	3.86e0	4.18e0	2.45e0	1.77e-16
Cholesky–QR (no pivoting)				1.23e-13
Cholesky–QR (with pivoting)				1.21e-16

displacement along the beam element, then the beam element stiffness matrix is [17]

$$K_i = \frac{2(EI)_i}{L_i^3} \begin{bmatrix} 6 & 3L_i & -6 & 3L_i \\ 3L_i & 2L_i^2 & -3L_i & L_i^2 \\ -6 & -3L_i & 6 & -3L_i \\ 3L_i & L_i^2 & -3L_i & 2L_i^2 \end{bmatrix}$$

and the beam element consistent mass matrix is

$$M_i = \frac{\overline{m}_i L_i}{420} \begin{bmatrix} 156 & 22L_i & 54 & -13L_i \\ 22L_i & 4L_i^2 & 13L_i & -3L_i^2 \\ 54 & 13L_i & 156 & -22L_i \\ -13L_i & -3L_i^2 & -22L_i & 4L_i^2 \end{bmatrix},$$

where $\overline{m}_i$ is the average mass per unit length for the ith beam. The global stiffness and mass inertia matrices are obtained by assembling the K_i and M_i, $i = 1\colon N$.

For our example, we chose $N = 5$ finite elements leading to 9 degrees of freedom and we varied the parameters e_i, L_i, $(EI)_i$, and $\overline{m}_i$, sometimes applying direct search to maximize the backward error over these variables. The backward errors for Algorithm 2.1 and the Cholesky–QR method with pivoting were always of order u, with our backward error bounds for Algorithm 2.1 also of order u. Table 5.2 shows results for two sets of parameters. The second set of results shows again that pivoting can be needed for stability of the Cholesky–QR method.

Example 3. This is an example where Algorithm 2.1 is unstable and there is only one large value of ω_k. With $n = 10$, we take $A \in \mathbb{R}^{n\times n}$ to be a random symmetric

TABLE 5.3
Iterative refinement of eigenpairs of Example 4. *For the entry marked* †, *convergence was not to the eigenvalue indicated in the leftmost column.*

	Before refinement		After refinement					
			Algorithm 4.1			Algorithm 4.2		
λ	$\eta_\infty(\widetilde{x},\widetilde{\lambda})$	$e(\widetilde{\lambda})$	$\eta_\infty(\widehat{x},\widehat{\lambda})$	$e(\widehat{\lambda})$	it	$\eta_\infty(\widehat{x},\widehat{\lambda})$	$e(\widehat{\lambda})$	it
$\epsilon = 2^{-6} \approx 1.6 \times 10^{-2}$								
1.4e0	4e-7	9e-6	5e-17	3e-16	2	7e-17	1e-15	2
−4.6e1	2e-8	6e-8	7e-18	2e-16	2	6e-18	2e-16	2
−8.4e3	2e-11	1e-9	5e-20	0	1	5e-20	0	2
$\epsilon = 2^{-8} \approx 3.9 \times 10^{-3}$								
1.4e0	2e-3	4e-2	4e-17	2e-16	3	4e-17	2e-16	12
−1.8e2	1e-5	3e-4	2e-17	2e-16	2	3e-17	8e-16	9
−1.4e4	4e-9	3e-6	5e-21	2e-16	2	2e-15	1e-12	∗
$\epsilon = 2^{-12} \approx 2.4 \times 10^{-4}$								
1.4e0	3e-3	1e0	4e-18	$0^\dagger$	5	1e-2	1e0	∗
−3.0e3	6e-4	8e-1	1e-22	0	5	3e-3	8e-1	∗
−3.5e7	4e-5	1e-1	2e-17	4e-16	3	2e-5	1e-1	∗

matrix and $B = I_n$ and replace the (n,n) entries of each matrix by 10^{-24}. Jacobi rotations not involving the nth plane have $\rho = 1$, and therefore ω_k is small. However, when we first apply a Jacobi rotation in the $(1,n)$ plane we see that $\rho = 10^{12}$ and

$$a_{11} - \rho^2 a_{nn} = a_{11} - 1 \ll \rho a_{1n} = 10^{12} a_{1n},$$

and therefore, from (3.6), $sc \approx 1/2$ and $\omega_k \approx 5 \times 10^{11}$. Note that this is an example where (3.30) is sharp. This is the only ill-conditioned M_k transformation as, using our scaling strategy, we set $\widetilde{d}_n^2 = c^2 d_n^2 + s^2 d_1^2 = O(1)$ in (3.17), and afterwards ρ is always approximately 1 for all subsequent rotations. The other key terms from the error bounds are $\max_k \pi_k = 8.4 \times 10^{11}$ and $\max_k \mu_k^2 = 2.0$. The computed eigenvalues consist of a group of 8 of order 1, all with backward errors of order 10^{-5} and two eigenvalues of order 10^{12}, with backward errors of order u. Applying Algorithm 4.1 to the eigenvalues with large backward errors we found that backward errors of order u were produced within 3–7 iterations; Algorithm 4.2 did not converge for any of the eigenvalues. The Cholesky–QR method was stable in this example.

Example 4. This example is one of a form suggested by G. W. Stewart that causes difficulties for Algorithm 2.1, and we use it to compare Algorithms 4.1 and 4.2. The matrices are

$$\operatorname{diag}(A) = d, \quad a_{ij} = \min(i,j) \text{ for } i \neq j, \quad B = \operatorname{diag}(d), \quad d = [1, \epsilon, \epsilon^2, \ldots, \epsilon^{n-1}]$$

with $0 < \epsilon < 1$. We take $n = 8$ with three choices of ϵ and concentrate on the three eigenvalues of smallest absolute value. We report in Table 5.3 the backward error $\eta_\infty(\widehat{x},\widehat{\lambda})$ of the computed eigenpair and the forward error

$$e(\widehat{\lambda}) = \frac{|\lambda - \widehat{\lambda}|}{|\lambda|}$$

of the computed eigenvalue, where the exact λ is obtained using MATLAB's Symbolic Math Toolbox; these statistics are given both before and after refinement, together

TABLE 5.4
Terms from error analysis for Example 4.

ϵ	$\kappa_2(B)$	$\max \omega_k$	$\max \mu_k^2$	$\max \pi_k$
2^{-6}	4e12	1.3e5	7.9	1.1e10
2^{-8}	7e16	1.7e7	8.0	8.8e13
2^{-12}	2e25	2.8e11	8.0	5.7e21

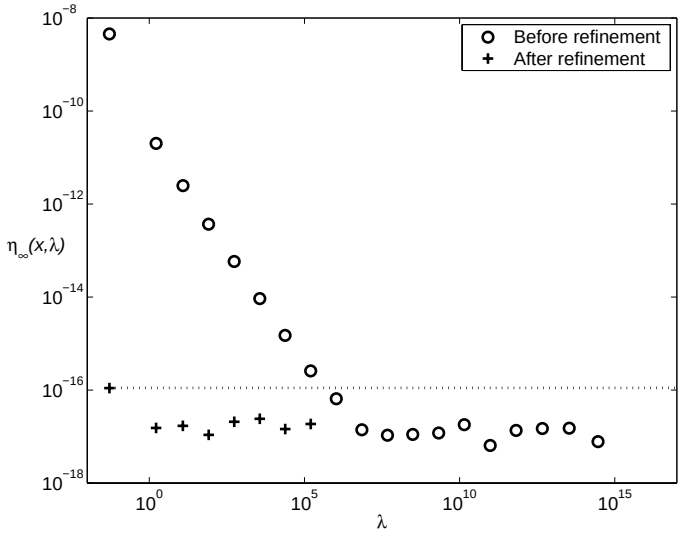


FIG. 5.2. *Backward errors for Cholesky–QR method before and after iterative refinement for Kahan matrix example (Example* 5*). Dotted line denotes unit roundoff level.*

with the number of iterations required by Algorithms 4.1 and 4.2, where "$*$" denotes no convergence after 50 iterations and in this case the quantities from the 50th iteration are shown. Table 5.4 shows the size of the terms appearing in the error bounds of section 3.3. The observed instability corresponds to large ω_k and π_k, but μ_k^2 is small, as is usually the case. We see that, as expected from the theory [23], refining with the unstable linear system solver produces the same limiting backward error as when the stable solver is used, but that it can produce slower convergence and is less likely to converge at all, as we saw also in Example 3. Iterative refinement also improves the forward error e. As one entry in the table shows, it is possible for iterative refinement to converge to a different eigenpair than expected when the original approximate eigenpair is sufficiently poor. The Cholesky–QR method performs stably on this example.

Example 5. The next example illustrates how ill condition of L can cause instability. Here, $n = 20$, $A = I$, and $B = R^T R$, where R is a Kahan matrix, and $\kappa_2(B) \approx 1/u$, $\kappa_2(L) \approx 3 \times 10^4$. Figure 5.2 plots the eigenvalues on the x-axis versus the ∞-norm backward errors of the eigenpairs on the y-axis, for eigenpairs both before and after refinement. At most one step of iterative refinement was required. The Cholesky–QR method was used, with Algorithm 4.2; Algorithms 2.1 and 4.1 give very similar results. The quantities in the error bounds for Algorithm 2.1 are $\max \omega_k = 0.6$, $\max \mu_k^2 = 315$, $\max \pi_k = 1.8$. As expected, it is the small eigenvalues that have large backward errors initially.

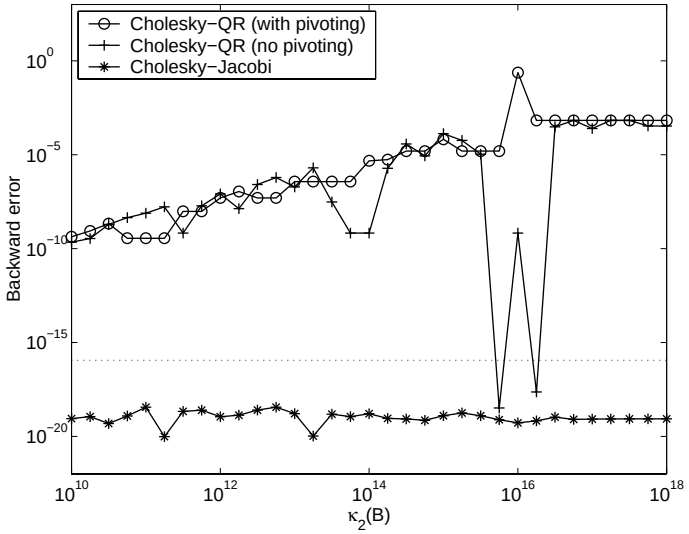


FIG. 5.3. *Behavior of the backward error for eigenvalue of smallest modulus of problem* (5.2) *with* $\alpha = 1$, $\delta = 10^{-3}$. *Dotted line denotes unit roundoff level.*

Example 6. Our penultimate example is adapted from a problem used by Fix and Heiberger [10] and shows that it is possible for the Cholesky–Jacobi method to be stable when the Cholesky–QR method both with and without pivoting is unstable. Let

$$A = \begin{bmatrix} 1 & \alpha & 0 & \delta \\ \alpha & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ \delta & 0 & 0 & \epsilon \end{bmatrix}, \quad B = \operatorname{diag}(\epsilon, 1, \epsilon, 1), \qquad \alpha, \delta > 0, \quad 0 < \epsilon < 1. \tag{5.2}$$

We solved the problem for $\alpha = 1$, $\delta = 10^{-3}$ and a range of ϵ from 10^{-10} to 10^{-18} by Algorithm 2.1 and the Cholesky–QR method with pivoting. Figure 5.3 plots the condition number of B against the backward error $\eta_2(\widehat{x}_{\min}, \widehat{\lambda}_{\min})$ for the eigenvalue $\widehat{\lambda}_{\min}$ of minimal modulus. The Cholesky–QR method performs unstably for most of the matrices B in the figure (strangely, producing generally better results without pivoting), while Algorithm 2.1 displays excellent stability. For Algorithm 2.1 we have $\max_k \omega_k = \max_k \pi_k = 1.0$ and $\max_k \mu_k^2 = 1.71$, so our error bounds predict the small backward errors.

Example 7. Our final example uses a class of random test problems suggested by Chandrasekaran [3]. They have the form

$$A = R + (10^{-8} n \lambda_n - \lambda_{[n/2]})I, \qquad B = S + (|\lambda_1| + 10^{-8} n \max(\lambda_1, \lambda_n))I,$$

where R and S are random matrices from the normal (0,1) distribution and $\lambda_1 \le \cdots \le \lambda_n$ are the eigenvalues of R (for A) or S (for B). With $5 \le n \le 100$ the backward errors of the eigenpairs produced by the Cholesky–Jacobi method and the Cholesky–QR method with and without pivoting were almost always less than nu, with a maximum value of 10^{-13} occurring for the Cholesky–QR method without pivoting for $n = 60$. Iterative refinement by Algorithms 4.1 and 4.2 reduced the backward error to u in at most three iterations, with only one iteration being required in over 95 percent of the cases. For Algorithm 2.1 we have $\max_k \omega_k = \max_k \mu_k^2 = 4$ and $\max_k \pi_k = 56$.

6. Conclusions. We have shown that the Cholesky–Jacobi method has better numerical stability properties than the standard backward error bound (1.7) suggests. For problems with an ill-conditioned B, the method can be, and often is, perfectly stable, and numerical experiments show that our bounds predict the stability well. The method is of practical use: it is easy to code, as Algorithm 2.1 shows, and the Jacobi method is particularly attractive in a parallel computing environment.

In practice, the Cholesky–QR method appears to perform as well as the Cholesky–Jacobi method, provided that complete pivoting is used in the Cholesky factorization. As we noted in section 1 this can, to some extent, be explained by the QR method's good performance on graded matrices. However, except for a rarely used variant employing Givens tridiagonalization, the best backward error bound for the Cholesky–QR method continues to contain a factor $\kappa_2(B)$. It is an important open problem to derive a sharper bound.

Instability of the Cholesky methods can be cured by iterative refinement, provided it is not too severe, as we have illustrated. Drawbacks are that refinement is expensive if applied to more than just a few eigenpairs, and practically verifiable conditions that guarantee convergence to the desired eigenpair are not available, though the method is surprisingly effective in practice.

The Cholesky–QR method (without pivoting) is the standard method for solving the symmetric definite generalized eigenproblem in LAPACK, MATLAB 6, and the NAG Library, all of which aim to provide exclusively backward stable algorithms. It is clearly desirable for these implementations to incorporate pivoting in the Cholesky factorization, in order to enhance the reliability, and to incorporate the option of iterative refinement of selected eigenpairs, to ameliorate those instances, which are rarer than we can explain, where the Cholesky–QR method behaves unstably.

Acknowledgment. We thank Sven Hammarling for many helpful discussions on this work.

REFERENCES

[1] E. Anderson, Z. Bai, C. H. Bischof, S. Blackford, J. W. Demmel, J. J. Dongarra, J. J. Du Croz, A. Greenbaum, S. J. Hammarling, A. McKenney, and D. C. Sorensen, *LAPACK Users' Guide*, 3rd ed., SIAM, Philadelphia, 1999.

[2] M. F. Anjos, S. J. Hammarling, and C. C. Paige, *Solving the Generalized Symmetric Eigenvalue Problem*, manuscript, 1992.

[3] S. Chandrasekaran, *An efficient and stable algorithm for the symmetric-definite generalized eigenvalue problem*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1202–1228.

[4] B. N. Datta, *Numerical Linear Algebra and Applications*, Brooks/Cole, Pacific Grove, CA, 1995.

[5] P. I. Davies, *Solving the Symmetric Definite Generalized Eigenvalue Problem*, Ph.D. thesis, University of Manchester, Manchester, England, 2000.

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

[7] J. W. Demmel and K. Veselić, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.

[8] J. J. Dongarra, *Algorithm* 589 *SICEDR: A FORTRAN subroutine for improving the accuracy of computed matrix eigenvalues*, ACM Trans. Math. Software, 8 (1982), pp. 371–375.

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

[10] G. Fix and R. Heiberger, *An algorithm for the ill-conditioned generalized eigenvalue problem*, SIAM J. Numer. Anal., 9 (1972), pp. 78–88.

[11] V. Frayssé and V. Toumazou, *A note on the normwise perturbation theory for the regular generalized eigenproblem*, Numer. Linear Algebra Appl., 5 (1998), pp. 1–10.

[12] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[13] D. J. Higham and N. J. Higham, *Structured backward error and condition of generalized eigenvalue problems*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 493–512.
[14] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
[15] N. J. Higham, *Iterative refinement for linear systems and LAPACK*, IMA J. Numer. Anal., 17 (1997), pp. 495–509.
[16] W. Kerner, *Large-scale complex eigenvalue problems*, J. Comput. Phys., 85 (1989), pp. 1–85.
[17] L. Meirovitch, *Elements of Vibration Analysis*, 2nd ed., McGraw-Hill, New York, 1986.
[18] C. B. Moler and G. W. Stewart, *An algorithm for generalized matrix eigenvalue problems*, SIAM J. Numer. Anal., 10 (1973), pp. 241–256.
[19] B. N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, 1997.
[20] G. Peters and J. H. Wilkinson, *$Ax = \lambda Bx$ and the generalized eigenproblem*, SIAM J. Numer. Anal., 7 (1970), pp. 479–492.
[21] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.
[22] H. J. Symm and J. H. Wilkinson, *Realistic error bounds for a simple eigenvalue and its associated eigenvector*, Numer. Math., 35 (1980), pp. 113–126.
[23] F. Tisseur, *Newton's method in floating point arithmetic and iterative refinement of generalized eigenvalue problems*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1038–1057.
[24] A. van der Sluis, *Condition numbers and equilibration of matrices*, Numer. Math., 14 (1969), pp. 14–23.
[25] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, UK, 1965.
[26] J. H. Wilkinson, *Error analysis revisited*, Bull. Inst. Math. Appl., 22 (1986), pp. 192–200.