SIAM J. MATRIX ANAL. APPL.
Vol. 21, No. 4, pp. 1202–1228

AN EFFICIENT AND STABLE ALGORITHM FOR THE SYMMETRIC-DEFINITE GENERALIZED EIGENVALUE PROBLEM*

S. CHANDRASEKARAN†

Abstract. A new, efficient, and stable algorithm for computing all the eigenvalues and eigenvectors of the problem $Ax = \lambda Bx$, where A is symmetric indefinite and B is symmetric positive definite, is proposed.

Key words. generalized eigenvalues, eigenvalues, eigenvectors, error analysis, deflation, perturbation bounds, pencils, stable algorithms

AMS subject classifications. 15A18, 15A22, 15A23, 15A42, 47A75, 65F15

PII. S0895479897316308

1. Introduction.

In this paper we consider the problem of computing the eigenvalues and eigenvectors of the pencil $Ax = \lambda Bx$, where A is a real symmetric-indefinite matrix and B is a real symmetric positive-definite matrix. Mathematically, this problem is equivalent to computing the eigendecomposition of the symmetric matrix $G^{-1}AG^{-T}$, where $B = GG^T$. Unfortunately, the approach is not numerically stable, but it does reveal some important properties about the eigenvalues and eigenvectors. First, all the eigenvalues must be real. Second, the eigenvector matrix diagonalizes both A and B simultaneously. In finite precision, the transformation $G^{-1}AG^{-T}$ leads to violation of the second property, while the QZ algorithm violates the first property.

In this paper we propose a new algorithm which satisfies both properties and is numerically stable and efficient.

Previous work on this problem, when the matrices are dense, has involved either trying to implement the transformation $G^{-1}AG^{-T}$ accurately or extending Jacobi, QZ, or other iterative type methods. See section 8.7 in [2], section 5.68 in [8], and chapter 15 in [6] for a summary of earlier work. Iterative methods, which can be used for both dense and sparse problems, have been studied more extensively. See [7] for a more extensive guide to the literature.

The outline of this paper is as follows. To convey the basic ideas we first outline the algorithm assuming that the symmetric eigenvalue problem can be solved exactly. We then point out the difficulties introduced by inexact calculations and the methods we propose for overcoming them. This is followed by an error analysis to prove the stability of the algorithm. We then discuss implementation issues and describe the experimental results which validate our claims. As part of the error analysis, we also establish a perturbation bound for the smallest eigenvalues in magnitude, which we believe to be new.

1.1. The key idea.

The problem can be viewed as the simultaneous LDL^T factorization of the matrices A and B, where L is now no longer constrained to be a triangular matrix. As is well known, while the LDL^T factorization of B (which is symmetric positive definite) is stable even without pivoting (Cholesky factorization),

*Received by the editors February 7, 1997; accepted for publication (in revised form) by F. T. Luk July 14, 1999; published electronically March 30, 2000. This research was supported in part by NSF grant CCR-9734290.

http://www.siam.org/journals/simax/21-4/31630.html

†Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 (shiv@ece.ucsb.edu).

the LDL^T factorization of A, which is symmetric indefinite, can be severely unstable without pivoting. When viewed from this framework, the problem is to discover a "permutation" (actually, any orthogonal transform) such that both A and B can be simultaneously factored into LDL^T, while assuring at the same time that stability is maintained in the factorization of A. This is a difficult task. The key idea in this paper is that the correct pivoting order is obtained when the eigenvalues are deflated in *decreasing order of magnitude*. This is an oversimplification, but it helps in understanding the rest of the paper.

2. Notation. We will follow Householder's convention [3] and denote matrices by capital Roman letters and vectors by small Roman letters. Scalars will be denoted by small Greek letters. Due to the necessity of keeping track of the various variables of the algorithm in the error analysis, we will avoid statements such as $A := A + I$. Instead the same statement will be denoted by $A_{;i+1} \leftarrow A_{;i} + I$. That is, the letters after the semicolon in the suffix help keep track of the position in the algorithm where the particular variable was created. The letters after the semicolon also give cues to where the particular variable was created. For example, $A_{;i;q}$ is used to indicate the matrix $A_{;i}$ after a similarity transformation by a matrix labeled by Q. Similarly the letters w, t, and s also appear in subscripts after semicolons and their cues can be inferred from their definition contexts. Also the notation $A_{2,3;i}$ denotes the element in the $(2,3)$ position of the matrix $A_{;i}$, and $A_{j;i}$ denotes the jth column of the matrix $A_{;i}$. We will use Matlab-type notation [5] for submatrices. Therefore, $A_{[i:n,p:q];i}$ denotes the submatrix of $A_{;i}$ formed from those elements in rows i to n which are also in columns p to q. When there is no necessity to refer to the elements of a matrix we will drop the semicolon in the suffix. This will be clear from the context. A bar over a variable indicates that it is a submatrix.

We use $\equiv$ when variables are being defined for the first time. In the description of algorithms we use left and right arrows to denote assignment. Therefore, $X_{;i} \leftarrow A_{;i}B_{;i}$ means that the product of the matrices $A_{;i}$ and $B_{;i}$ is assigned to the variable $X_{;i}$. Whereas,

$$A_{;i}B_{;i} \rightarrow U_{;i}\Sigma_{;i}V_{;i}^T, \qquad \text{compute SVD},$$

means that the SVD of the product $A_{;i}B_{;i}$ is computed, and $U_{;i}$ is assigned the left singular vector matrix, $\Sigma_{;i}$ is assigned the matrix of singular values, and $V_{;i}$ is assigned the right singular vector matrix.

3. In infinite precision. In this section we first present the algorithm assuming that all calculations (including some eigendecompositions!) can be done exactly. This is to enable us to present the basic ideas in an uncluttered manner.

We assume that A is nonsingular. If it is not, then the zero eigenvalues can be easily deflated from the problem, as detailed in section 5.

We note that the main difficulty in finding the eigenvalues and eigenvectors is due to finite precision effects. So the objective is to try to design an algorithm which will work for problems, where the symmetric positive-definite matrix B is almost numerically singular. For otherwise, if the matrix B is well-conditioned, we can compute the transformation $G^{-1}AG^{-T}$ with little loss of accuracy.

One key observation we make now is that even when both A and B are highly ill-conditioned we can compute the matrix $G^TA^{-1}G$ to sufficient accuracy so as to enable us to compute its largest eigenvalues in magnitude to backward accuracy. One way to proceed now is to deflate these computed eigenvalues from A and B and to

work recursively on the smaller pencil. Unfortunately, the deflation requires transformations whose cumulative condition number is as high as the condition number of B. So we seem to fare no better.

This is where our next key observation comes in: if we deflate the eigenvalues of $G^T A^{-1} G$ in the order of decreasing size in magnitude, then we can ensure that the resulting sequence of deflating transformations *can* be implemented in a numerically stable manner. The rest of this paper is devoted to showing why and how this can be done.

We first begin by showing why it is necessary to deflate the eigenvalues in the order of decreasing size in magnitude. For that purpose we present a version of the algorithm here which assumes that all computations can be done exactly, including some eigenvalue decompositions.

ALGORITHM I. USING EXACT EIGENDECOMPOSITIONS.

Begin

1. $A_{;1} \equiv A; \qquad B \equiv B_{;1} \rightarrow U \Sigma U^T$; compute eigendecomposition of B.
2. $\sqrt{\Sigma} U^T A_{;1}^{-1} U \sqrt{\Sigma} \rightarrow V \Lambda V^T$; compute the eigendecomposition of equivalent symmetric matrix such that the eigenvalues are ordered from largest to smallest in magnitude.
3. $X_{;1} \equiv X \leftarrow A_{;1}^{-1} U \sqrt{\Sigma} V$; compute the generalized eigenvectors of the pencil.
4. We now deflate the eigenvectors from the pencil.
   For $i = 1$ **to** n **do**
   (a) Compute Householder transform $Q_{;i}$ such that $Q_{;i} X_{i;i}$ is parallel to e_i.
   (b) $A_{;i;q} \leftarrow Q_{;i} A_{;i} Q_{;i}^T, \quad B_{;i;q} \leftarrow Q_{;i} B_{;i} Q_{;i}^T, \quad X_{;i;q} \leftarrow Q_{;i} X_{;i}$.
   (c) Compute Householder transform W_i such that $W_i A_{i;i;q}$ has zeros below the $(i+1)$st component. It follows that $W_i B_{i;i;q}$ also has zeros below the $(i+1)$st component.
   (d) $A_{;i;w} \leftarrow W_i A_{;i;q} W_i^T, \quad B_{;i;w} \leftarrow W_i B_{;i;q} W_i^T, \quad X_{;i;w} \leftarrow W_i X_{;i;q}$.
   (e) Note that $X_{i;i;w}$ is still parallel to e_i. Therefore, $A_{i;i;w}$ and $B_{i;i;w}$ are parallel and in the span of e_i and e_{i+1}. Therefore, we can find one elementary Gauss transform L_i^{-1} such that $L_i^{-1} A_{i;i;w}$ and $L_i^{-1} B_{i;i;w}$ are both parallel to e_i.
   (f) $A_{;i+1} \leftarrow L_i^{-1} A_{;i;w} L_i^{-T}, \quad B_{;i+1} \leftarrow L_i^{-1} B_{;i;w} L_i^{-T}, \quad X_{;i+1} \leftarrow L_i^T X_{;i;w}$.
   endfor
5. We now have

$$(L_n^{-1} W_n Q_{;n}) \cdots (L_1^{-1} W_1 Q_{;1}) A ((L_n^{-1} W_n Q_{;n}) \cdots (L_1^{-1} W_1 Q_{;1}))^T \equiv D_A,$$
$$(L_n^{-1} W_n Q_{;n}) \cdots (L_1^{-1} W_1 Q_{;1}) B ((L_n^{-1} W_n Q_{;n}) \cdots (L_1^{-1} W_1 Q_{;1}))^T \equiv D_B,$$

where D_A and D_B are diagonal matrices. Now define

$$C \equiv Q_{;1}^T W_1^T L_1 \cdots Q_{;n}^T W_n^T L_n. \tag{1}$$

Then we have $C D_A C^T = A$ and $C D_B C^T = B$, where D_A and D_B are diagonal matrices. Therefore, the generalized eigenvalues can be obtained as the ratios of the diagonal elements of D_A and D_B, and the generalized eigenvectors can be obtained from the inverse of C by using the factored form (1).

End

We now look in more detail at the transforms $Q_{;i}$, W_i, and L_i used during the deflation process. We claim that

$$X_{;i} \equiv \begin{array}{c} \\ i-1 \\ n-i+1 \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} D_{;X;i} & 0 & 0 \\ 0 & x_i & \bar{X}_{;i} \end{pmatrix} \end{array}, \tag{2}$$

$$A_{;i} \equiv \begin{array}{c} \\ i-1 \\ n-i+1 \end{array} \begin{array}{c} \begin{array}{cc} i-1 & n-i+1 \end{array} \\ \begin{pmatrix} \bar{D}_{;A;i} & 0 \\ 0 & \bar{A}_{;i} \end{pmatrix} \end{array}, \tag{3}$$

$$B_{;i} \equiv \begin{array}{c} \\ i-1 \\ n-i+1 \end{array} \begin{array}{c} \begin{array}{cc} i-1 & n-i+1 \end{array} \\ \begin{pmatrix} \bar{D}_{;B;i} & 0 \\ 0 & \bar{B}_{;i} \end{pmatrix} \end{array}, \tag{4}$$

where D's denote diagonal matrices. These facts will be proved by induction. It is obviously true for $i = 1$. Invoking the induction hypothesis, we see that

$$Q_{;i} \equiv \begin{array}{c} \\ i-1 \\ n-i+1 \end{array} \begin{array}{c} \begin{array}{cc} i-1 & n-i+1 \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & \bar{Q}_i \end{pmatrix} \end{array}, \qquad \bar{Q}_i x_i = \pm \|x_i\| e_i,$$

$$X_{;i;q} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} D_{;X;i} & 0 & 0 \\ 0 & \pm\|x_i\| & h_i^T \\ 0 & 0 & \bar{X}_{;i;q} \end{pmatrix} \end{array},$$

$$A_{;i;q} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;A;i} & 0 & 0 \\ 0 & \alpha_i & a_{;i;q}^T \\ 0 & a_{;i;q} & \bar{A}_{;i;q} \end{pmatrix} \end{array},$$

$$B_{;i;q} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;B;i} & 0 & 0 \\ 0 & \beta_i & b_{;i;q}^T \\ 0 & b_{;i;q} & \bar{B}_{;i;q} \end{pmatrix} \end{array}, \qquad \lambda_i a_{;i;q} = b_{;i;q},$$

where $\Lambda \equiv \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$. Therefore,

$$W_i \equiv \begin{array}{c} \\ i \\ n-i \end{array} \begin{array}{c} \begin{array}{cc} i & n-i \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & \bar{W}_i \end{pmatrix} \end{array}, \qquad \bar{W}_i a_{;i;q} = \pm \|a_{;i;q}\| e_i,$$

$$X_{;i;w} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} D_{;X;i} & 0 & 0 \\ 0 & \pm\|x_i\| & h_i^T \\ 0 & 0 & \bar{X}_{;i;w} \end{pmatrix} \end{array},$$

$$A_{;i;w} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;A;i} & 0 & 0 \\ 0 & \alpha_i & \pm\|a_{;i;q}\| e_1^T \\ 0 & \pm\|a_{;i;q}\| e_1 & \bar{A}_{;i;w} \end{pmatrix} \end{array},$$

$$B_{;i;w} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;B;i} & 0 & 0 \\ 0 & \beta_i & \pm\|b_{;i;q}\|e_1^T \\ 0 & \pm\|b_{;i;q}\|e_1 & \bar{B}_{;i;w} \end{pmatrix} \end{array},$$

$$\begin{pmatrix} \beta_i \\ \pm\|b_{;i;q}\| \end{pmatrix} = \lambda_i \begin{pmatrix} \alpha_i \\ \pm\|a_{;i;q}\| \end{pmatrix}.$$

Therefore, we have that

$$L_i^{-1} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \rho_i e_1 & I \end{pmatrix} \end{array}, \qquad \rho_i = -\frac{\pm\|b_{;i;q}\|}{\beta_i} = -\frac{\pm\|a_{;i;q}\|}{\alpha_i},$$

$$X_{;i+1} = \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} D_{;X;i} & 0 & 0 \\ 0 & \pm\|x_i\| & 0 \\ 0 & 0 & \bar{X}_{;i;w} \end{pmatrix} \end{array},$$

$$A_{;i+1} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;A;i} & 0 & 0 \\ 0 & \alpha_i & 0 \\ 0 & 0 & \bar{A}_{;i;w} - \frac{\|a_{;i;q}\|^2}{\alpha_i} e_1 e_1^T \end{pmatrix} \end{array},$$

$$B_{;i+1} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;B;i} & 0 & 0 \\ 0 & \beta_i & 0 \\ 0 & 0 & \bar{B}_{;i;w} - \frac{\|b_{;i;q}\|^2}{\beta_i} e_1 e_1^T \end{pmatrix} \end{array},$$

where the structure of the ith row of $X_{;i+1}$ is obtained by looking at the form of $A_{;i+1}$ and $B_{;i+1}$. This completes our induction and proves the structures assumed in (2), (3), and (4).

We now show that the norms of the Schur complements of A generated by the transforms L_i grow no faster than those which occur in Gaussian elimination with partial pivoting. This is one of the reasons why our approach leads to a numerically stable algorithm.

Define the following two submatrices of $A_{;i;w}$ and $B_{;i;w}$:

$$\bar{A}_{;i;l} \equiv \begin{array}{c} \\ 1 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 1 & 1 \end{array} \\ \begin{pmatrix} \alpha_i & \pm\|a_{;i;q}\| \\ \pm\|a_{;i;q}\| & \bar{A}_{1,1;i;w} \end{pmatrix} \end{array},$$

$$\bar{B}_{;i;l} \equiv \begin{array}{c} \\ 1 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 1 & 1 \end{array} \\ \begin{pmatrix} \beta_i & \pm\|b_{;i;q}\| \\ \pm\|b_{;i;q}\| & \bar{B}_{1,1;i;w} \end{pmatrix} \end{array} = \begin{pmatrix} \lambda_i\alpha_i & \pm\|a_{;i;q}\|\lambda_i \\ \pm\|a_{;i;q}\|\lambda_i & \bar{B}_{1,1;i;w} \end{pmatrix}.$$

Note that $\bar{B}_{;i;l}$ is symmetric positive definite. From this we get, using determinants, that

$$\bar{B}_{1,1;i;w}\lambda_i\alpha_i > \|a_{;i;q}\|^2\lambda_i^2,$$

which implies that

$$|\lambda_i| < \bar{B}_{1,1;i;w} \frac{|\alpha_i|}{\|a_{;i;q}\|^2}. \tag{5}$$

Since we required the following ordering of the eigenvalues

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|,$$

it follows that λ_i is also the largest eigenvalue in magnitude of the pencil $\lambda \bar{A}_{;i;l} x = \bar{B}_{;i;l}$. Using the variational characterization of eigenvalues, we get

$$|\lambda_i| > \frac{\bar{B}_{1,1;i;w}}{|\bar{A}_{1,1;i;w}|}. \tag{6}$$

From inequalities (5) and (6) we get

$$|\bar{A}_{1,1;i;w}| > \frac{\|a_{;i;q}\|^2}{|\alpha_i|}.$$

Therefore,

$$\left| \bar{A}_{1,1;i;w} - \frac{\|a_{;i;q}\|^2}{\alpha_i} \right| < 2|\bar{A}_{1,1;i;w}|. \tag{7}$$

This indicates that the elements of D_A can grow at most like 2^n. Also,

$$\sqrt{|\alpha_i|}|\rho_i| = \sqrt{\frac{\|a_{;i;q}\|^2}{|\alpha_i|}} < \sqrt{|\bar{A}_{1,1;i;w}|}.$$

These facts by themselves are not sufficient to establish the numerical stability of the algorithm. We now proceed to look at the effects of errors in the eigendecomposition computation.

4. Error propagation. In this section we assume that $\|B\| = \|A\| = 1$. We consider the effects of the truncation error

$$G \equiv U\sqrt{\Sigma}, \qquad G^T A^{-1} G \to \hat{V}\hat{\Lambda}\hat{V}^T + E,$$

where $\|E\| \leq \epsilon\|\hat{\Lambda}\|$. Recovering the generalized eigenvector, we have that

$$GG^T(A^{-1}G\hat{v}_i) + GE\hat{v}_i = \hat{\lambda}_i A(A^{-1}G\hat{v}_i).$$

Define $\hat{x}_i \equiv A^{-1}G\hat{v}_i$ and rearrange the above expression to get it in normalized backward error form:

$$\begin{aligned} GG^T \frac{\hat{x}_i}{\|\hat{x}_i\|} + \frac{GE\hat{v}_i}{\|\hat{x}_i\|} &= \hat{\lambda}_i A \frac{\hat{x}_i}{\|\hat{x}_i\|} \qquad \text{if } |\hat{\lambda}_i| < 1, \\ \frac{1}{\hat{\lambda}_i} GG^T \frac{\hat{x}_i}{\|\hat{x}_i\|} + \frac{GE\hat{v}_i}{\hat{\lambda}_i\|\hat{x}_i\|} &= A \frac{\hat{x}_i}{\|\hat{x}_i\|} \qquad \text{if } |\hat{\lambda}_i| \geq 1. \end{aligned}$$

From the above two equations it is clear that not all computed eigenpairs, $(\hat{\lambda}_i, \hat{x}_i)$, will be sufficiently accurate, and possibly no eigenpair is exact.

We modify the algorithm to take care of these possibilities. The new variables in the modified algorithm will have a "$;t$" in their suffix to distinguish them from similar variables occurring in Algorithm I.

ALGORITHM II. USING INEXACT EIGENDECOMPOSITIONS.

Begin

1. Assume $\|A\| = \|B\|$; else rescale A and B.
2. Assume A is nonsingular.
3. $A \equiv A_{;1;t}$; $\quad B \equiv B_{;1;t} \rightarrow U_{;1;t}\Sigma_{;1;t}U_{;1;t}^T$; compute eigendecomposition of B.
4. $\sqrt{\Sigma_{;1;t}}U_{;1;t}^T A_{;1;t}^{-1}U_{;1;t}\sqrt{\Sigma_{;1;t}} \rightarrow V_{;1;t}\Lambda_{;1;t}V_{;1;t}^T + E_{;1;t}$; compute the eigendecomposition of the equivalent symmetric matrix such that the eigenvalues are ordered from largest to smallest in magnitude.
5. $X_{;1;t} \leftarrow A_{;1;t}^{-1}U_{;1;t}\sqrt{\Sigma_{;1;t}}V_{;1;t}$; compute the generalized eigenvectors of the pencil.
6. $i \leftarrow 1$; we now deflate the eigenvectors from the pencil.
   While $(i < n)$ **do**
   (a) **While** $(i < n)$ **and**
   $(\|(\lambda_{i;i;t}A_{;i;t} - B_{;i;t})X_{i;i;t}\| \leq \epsilon\|X_{i;i;t}\|(|\lambda_{i;i;t}|\ \|A_{[i:n],[i:n];i;t}\| + \|B_{[i:n],[i:n];i;t}\|)$
   do
   (i) Compute Householder transform $Q_{;i;t}$ such that $Q_{;i;t}X_{i;i;t}$ is parallel to e_i.
   (ii) $A_{;i;q;t} \leftarrow Q_{;i;t}A_{;i;t}Q_{;i;t}^T$, $\quad B_{;i;q;t} \leftarrow Q_{;i;t}B_{;i;t}Q_{;i;t}^T$, $\quad X_{;i;q;t} \leftarrow Q_{;i;t}X_{;i;t}$
   (iii) **If** $|\lambda_{i;i;t}| \geq 1$, **then**
   A. $(A_{i;i;q;t} \leftarrow B_{i;i;q;t}/\lambda_{i;i;t};\ A_{i,[1:n];i;q;t} \leftarrow A_{i;i;q;t}^T) \rightarrow A_{;i;p;t}$; make ith column and row of $A_{;i;q;t}$ exactly parallel to that of $B_{;i;q;t}$.
   B. Compute Householder transform $W_{i;t}$ such that $W_{i;t}B_{i;i;q;t}$ has zeros below the $(i+1)$st component.
   C. $A_{;i;w;t} \leftarrow W_{i;t}A_{;i;p;t}W_i^T$, $\quad B_{;i;w;t} \leftarrow W_{i;t}B_{;i;q;t}W_{i;t}^T$, $\quad X_{;i;w;t} \leftarrow W_{i;t}X_{;i;q;t}$.
   D. Find an elementary Gauss transform $L_{;i;t}^{-1}$ such that $L_{;i;t}^{-1}B_{i;i;w;t}$ is parallel to e_i.
   E. $A_{;i+1;t} \leftarrow L_{;i;t}^{-1}A_{;i;w;t}L_{;i;t}^{-T}$, $\ B_{;i+1;t} \leftarrow L_{;i;t}^{-1}B_{;i;w;t}L_{;i;t}^{-T}$, $\ X_{;i+1;t} \leftarrow L_{;i;t}^T X_{;i;w;t}$.
   (iv) **else**; $(|\lambda_{i;i;t}| < 1)$
   A. $(B_{i;i;q;t} \leftarrow \lambda_{i;i;t}A_{i;i;q;t};\ B_{i,[1:n];i;q;t} \leftarrow B_{i;i;q;t}^T) \rightarrow B_{;i;p;t}$; make ith column and row of $B_{;i;q;t}$ exactly parallel to that of $A_{;i;q;t}$.
   B. Compute Householder transform $W_{i;t}$ such that $W_{i;t}A_{i;i;q;t}$ has zeros below the $(i+1)$st component.
   C. $B_{;i;w;t} \leftarrow W_{i;t}B_{;i;p;t}W_i^T$, $\quad A_{;i;w;t} \leftarrow W_{i;t}A_{;i;q;t}W_{i;t}^T$, $\quad X_{;i;w;t} \leftarrow W_{i;t}X_{;i;q;t}$.
   D. Find an elementary Gauss transform $L_{;i;t}^{-1}$ such that $L_{;i;t}^{-1}A_{i;i;w;t}$ is parallel to e_i.
   E. $B_{;i+1;t} \leftarrow L_{;i;t}^{-1}B_{;i;w;t}L_{;i;t}^{-T}$, $\ A_{;i+1;t} \leftarrow L_{;i;t}^{-1}A_{;i;w;t}L_{;i;t}^{-T}$, $\ X_{;i+1;t} \leftarrow L_{;i;t}^T X_{;i;w;t}$.
   endif
   (v) $\Lambda_{;i+1;t} \leftarrow \Lambda_{;i;t}$.
   (vi) $i \leftarrow i + 1$.
   endwhile
   (b) **If** $(i < n)$, **then**
   (i) $B_{;i;t} \rightarrow U_{;i;t}\Sigma_{;i;t}U_{;i;t}^T$; recompute eigendecomposition of $B_{;i;t}$.
   (ii) $\sqrt{\Sigma_{;i;t}}U_{;i;t}^T A_{;i;t}^{-1}U_{;i;t}\sqrt{\Sigma_{;i;t}} \rightarrow V_{;i;t}\Lambda_{;i;t}V_{;i;t}^T + E_{;i;t}$; recompute the eigendecomposition of equivalent symmetric matrix corresponding to the lower-right $(n-i+1) \times (n-i+1)$ block, such that the eigenvalues

of the submatrix are ordered from largest to smallest in magnitude.

(iii) $X_{:i:t} \leftarrow A_{:i:t}^{-1} U_{:i:t} \sqrt{\Sigma_{:i:t}} V_{:i:t}$; recompute the generalized eigenvectors of the pencil.

endif

endwhile

7. We now have the two diagonal matrices

$$D_{A:t} \equiv A_{:n:t}, \qquad D_{B:t} \equiv B_{:n:t},$$

and the accumulated transform

$$C_{:t} \equiv Q_{:1:t}^T W_{:1:t}^T L_{:1:t} \cdots Q_{:n:t}^T W_{:n:t}^T L_{:n:t},$$

such that

$$C_{:t} D_{B:t} C_{:t}^T = B + E_{:B:t},$$

$$\|E_{:B:t}\| \leq 2n\epsilon(\|A\| + n\|B\|) + O(\epsilon^2), \tag{8}$$

$$\frac{\|(D_{i,i;B;t} A - D_{i,i;A;t} B) C_{:t}^{-T} e_i\|}{\|C_{:t}^{-T} e_i\| \, (|D_{i,i;B;t}| \, \|A\| + |D_{i,i;A;t}| \, \|B\|)} \leq 5n^2\epsilon + O(\epsilon^2). \tag{9}$$

End

The important difference between Algorithm I and Algorithm II is that the latter deflates eigenvectors only after checking that they are sufficiently accurate, and if they are not it recomputes them.

Inequality (9) establishes that each computed eigenpair is computed to backward accuracy. Inequality (8) establishes that all the eigenpairs have been computed, in the sense that the eigenvector matrix diagonalizes B to backward accuracy. We now prove the claims in step 7 of Algorithm II.

We first claim that

$$A_{:i:t} \equiv \begin{array}{c} \\ i-1 \\ n-i+1 \end{array} \begin{array}{c} \begin{array}{cc} i-1 & n-i+1 \end{array} \\ \begin{pmatrix} \bar{D}_{:A:i:t} & 0 \\ 0 & \bar{A}_{:i:t} \end{pmatrix} \end{array}, \tag{10}$$

$$B_{:i:t} \equiv \begin{array}{c} \\ i-1 \\ n-i+1 \end{array} \begin{array}{c} \begin{array}{cc} i-1 & n-i+1 \end{array} \\ \begin{pmatrix} \bar{D}_{:B:i:t} & 0 \\ 0 & \bar{B}_{:i:t} \end{pmatrix} \end{array}. \tag{11}$$

These facts will be proved by induction. It is obviously true for $i = 1$. For general i, two cases are possible: either $X_{:i:t}$ has been computed in step 6(b)(iii) (or step 5) of Algorithm II or we have successfully passed the test of the while loop in step 6(a) and are entering step 6(a)(i).

We first consider the case when $X_{:i:t}$ is being computed in step 6(b) (or step 5). Since this is obviously true for $i = 1$, we can invoke the induction hypothesis. We now establish that we will pass the test

$$\|(\lambda_{i:i:t} A_{:i:t} - B_{:i:t}) X_{i:i:t}\| \leq \epsilon \|X_{i:i:t}\| (|\lambda_{i:i:t}| \, \|A_{[i:n],[i:n];i:t}\| + \|B_{[i:n],[i:n];i:t}\|) \tag{12}$$

in step 6(a). Using the induction hypothesis we can conclude that

$$U_{:i:t} \equiv \begin{array}{c} \\ i-1 \\ n-i+1 \end{array} \begin{array}{c} \begin{array}{cc} i-1 & n-i+1 \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & \bar{U}_{:i:t} \end{pmatrix} \end{array},$$

$$\Sigma_{;i;t} \equiv \begin{array}{c} \\ i-1 \\ n-i+1 \end{array} \begin{array}{c} \begin{array}{cc} i-1 & n-i+1 \end{array} \\ \begin{pmatrix} \bar{\Sigma}_{;i;u;t} & 0 \\ 0 & \bar{\Sigma}_{;i;l;t} \end{pmatrix} \end{array},$$

such that $\bar{B}_{;i;t} = \bar{U}_{;i;t}\bar{\Sigma}_{;i;l;t}\bar{U}^T_{;i;t}$. Therefore, we have

$$V_{;i;t} \equiv \begin{array}{c} \\ i-1 \\ n-i+1 \end{array} \begin{array}{c} \begin{array}{cc} i-1 & n-i+1 \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & \bar{V}_{;i;t} \end{pmatrix} \end{array},$$

$$\Lambda_{;i;t} \equiv \begin{array}{c} \\ i-1 \\ n-i+1 \end{array} \begin{array}{c} \begin{array}{cc} i-1 & n-i+1 \end{array} \\ \begin{pmatrix} \bar{\Lambda}_{;i;u;t} & 0 \\ 0 & \bar{\Lambda}_{;i;l;t} \end{pmatrix} \end{array},$$

where we assume that

$$\left\| \sqrt{\bar{\Sigma}_{;i;l;t}}\bar{U}^T_{;i;t}\bar{A}^{-1}_{;i;t}\bar{U}_{;i;t}\sqrt{\bar{\Sigma}_{;i;l;t}} - \bar{V}_{;i;t}\bar{\Lambda}_{;i;l;t}\bar{V}^T_{;i;t} \right\| = \|\bar{E}_{;i;t}\| \leq \epsilon \|\bar{\Lambda}_{;i;l;t}\|. \tag{13}$$

Therefore, this establishes that in step 6(b)(iii) we will have

$$X_{;i;t} \equiv \begin{array}{c} \\ i-1 \\ n-i+1 \end{array} \begin{array}{c} \begin{array}{cc} i-1 & n-i+1 \end{array} \\ \begin{pmatrix} D_{;X;i;t} & 0 \\ 0 & \bar{X}_{;i;t} \end{pmatrix} \end{array}.$$

Therefore, the test (12) can be rewritten as

$$\|(\lambda_{i;i;t}\bar{A}_{;i;t} - \bar{B}_{;i;t})\bar{X}_{i;i;t}\| \leq \epsilon \|\bar{X}_{;i;t}\|(|\lambda_{i;i;t}|\ \|\bar{A}_{;i;t}\| + \|\bar{B}_{;i;t}\|). \tag{14}$$

From (13) we have that

$$\begin{aligned} \bar{B}_{;i;t}\left(\bar{A}^{-1}_{;i;t}\bar{U}_{;i;t}\sqrt{\bar{\Sigma}_{;i;l;t}}\bar{V}_{j;i;t}\right) + \bar{U}_{;i;t}\sqrt{\bar{\Sigma}_{;i;l;t}}\bar{E}_{;i;t}\bar{V}_{j;i;t} \\ = \lambda_{j;i;l;t}\bar{A}_{;i;t}\left(\bar{A}^{-1}_{;i;t}\bar{U}_{;i;t}\sqrt{\bar{\Sigma}_{;i;l;t}}\bar{V}_{j;i;t}\right), \qquad j = 1, \ldots, n-i+1, \end{aligned}$$

or

$$\bar{B}_{;i;t}\bar{X}_{j;i;t} + \bar{U}_{;i;t}\sqrt{\bar{\Sigma}_{;i;l;t}}\bar{E}_{;i;t}\bar{V}_{j;i;t} = \lambda_{j;i;l;t}\bar{A}_{;i;t}\bar{X}_{j;i;t}, \quad j = 1, \ldots, n-i+1. \tag{15}$$

Since $\|\bar{\Lambda}_{;i;l;t}\| = |\lambda_{1;i;l;t}|$, we have that

$$\begin{aligned} |\lambda_{1;i;l;t}|(1-\epsilon) &\leq \left\|\bar{U}_{;i;t}\sqrt{\bar{\Sigma}_{;i;l;t}}\right\| \ \left\|\bar{A}^{-1}_{;i;t}\bar{U}_{;i;t}\sqrt{\bar{\Sigma}_{;i;l;t}}\bar{V}_{1;i;t}\right\| \\ &= \left\|\bar{U}_{;i;t}\sqrt{\bar{\Sigma}_{;i;l;t}}\right\| \ \left\|\bar{X}_{1;i;t}\right\|. \end{aligned}$$

Therefore,

$$\frac{\|\bar{U}_{;i;t}\sqrt{\bar{\Sigma}_{;i;l;t}}\bar{E}_{;i;t}\bar{V}_{1;i;t}\|}{\|\bar{X}_{1;i;t}\|} \leq \frac{\epsilon}{1-\epsilon}\|\bar{\Sigma}_{;i;l;t}\|. \tag{16}$$

Using arguments similar to the derivation of (6), we can show that

$$|\lambda_{1;i;l;t}|(1+\epsilon) \geq \frac{\|\bar{B}_{;i;t}\|}{\|\bar{A}_{;i;t}\|}.$$

Hence it follows that

$$|\lambda_{1;i;l;t}|\|\bar{A}_{;i;t}\| + \|\bar{B}_{;i;t}\| \geq \|\bar{B}_{;i;t}\|\frac{2+\epsilon}{1+\epsilon}. \tag{17}$$

Since

$$\frac{2+\epsilon}{1+\epsilon} \geq \frac{1}{1-\epsilon},$$

for sufficiently small ϵ, it follows from (15), (16), and (17) that the test (14) will be passed by at least $\bar{X}_{1;i;t}$.

Now we go ahead and analyze step 6(a) of Algorithm II. By the induction hypothesis and the preceding discussion, it is clear that

$$Q_{;i;t} \equiv \begin{array}{c} \\ i-1 \\ n-i+1 \end{array} \begin{array}{c} \begin{array}{cc} i-1 & n-i+1 \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & \bar{Q}_{i;t} \end{pmatrix} \end{array}, \qquad \bar{Q}_i X_{i;i;t} = \pm\|X_{i;i;t}\|e_i, \tag{18}$$

$$X_{;i;q;t} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} D_{;X;i;t} & 0 & 0 \\ 0 & \pm\|X_{i;i;t}\| & h_{i;t}^T \\ 0 & 0 & \bar{X}_{;i;q;t} \end{pmatrix} \end{array},$$

$$A_{;i;q;t} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;A;i;t} & 0 & 0 \\ 0 & \alpha_{i;t} & a_{;i;q;t}^T \\ 0 & a_{;i;q;t} & \bar{A}_{;i;q;t} \end{pmatrix} \end{array}, \tag{19}$$

$$B_{;i;q;t} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;B;i;t} & 0 & 0 \\ 0 & \beta_{i;t} & b_{;i;q;t}^T \\ 0 & b_{;i;q;t} & \bar{B}_{;i;q;t} \end{pmatrix} \end{array}, \tag{20}$$

where we now have that

$$\left\| \lambda_{i;i;t} \begin{pmatrix} \alpha_{i;t} \\ a_{;i;q;t} \end{pmatrix} - \begin{pmatrix} \beta_{i;t} \\ b_{;i;q;t} \end{pmatrix} \right\| \leq \epsilon(|\lambda_{i;i;t}|\ \|\bar{A}_{;i;t}\| + \|\bar{B}_{;i;t}\|). \tag{21}$$

That is, $(\,\alpha_{i;t} \quad a_{;i;q;t}^T\,)^T$ and $(\,\beta_{i;t} \quad b_{;i;q;t}^T\,)^T$ are no longer perfectly parallel.

At this stage the algorithm can follow two paths, depending on $\lambda_{i;i;t}$ (step 6(a)(iii)). We will first follow the path taken when $|\lambda_{i;i;t}| \geq 1$.

In step 6(a)(iii)(A) we perturb $A_{;i;q;t}$ such that its ith column is parallel to that of $B_{;i;q;t}$. Hence we have that

$$A_{;i;p;t} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;A;i;t} & 0 & 0 \\ 0 & \beta_{i;t}/\lambda_{i;i;t} & b_{;i;q;t}^T/\lambda_{i;i;t} \\ 0 & b_{;i;q;t}/\lambda_{i;i;t} & \bar{A}_{;i;q;t} \end{pmatrix} \end{array}. \tag{22}$$

Since $|\lambda_{i;i;t}| \geq 1$, it follows from (21) that

$$\|A_{;i;q;t} - A_{;i;p;t}\| \leq \epsilon(\|\bar{A}_{;i;t}\| + \|\bar{B}_{;i;t}\|). \tag{23}$$

The remaining steps in 6(a)(iii) are similar to Algorithm I and we have accordingly that

$$W_{;i;t} \equiv \begin{array}{c} \\ i \\ n-i \end{array} \begin{array}{c} \begin{array}{cc} i & n-i \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & \bar{W}_{;i;t} \end{pmatrix} \end{array}, \qquad \bar{W}_i b_{;i;q;t} = \pm\|b_{;i;q;t}\| e_i, \tag{24}$$

$$X_{;i;w;t} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} D_{;X;i;t} & 0 & 0 \\ 0 & \pm\|X_{i;i;t}\| & h_{;i;t}^T \\ 0 & 0 & \bar{X}_{;i;w;t} \end{pmatrix} \end{array},$$

$$A_{;i;w;1;t} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;A;i;t} & 0 & 0 \\ 0 & \beta_{i;t}/\lambda_{i;i;t} & \pm\|b_{;i;q;t}\| e_1^T/\lambda_{i;i;t} \\ 0 & \pm\|b_{;i;q;t}\| e_1/\lambda_{i;i;t} & \bar{A}_{;i;w;t} \end{pmatrix} \end{array},$$

$$B_{;i;w;t} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;B;i;t} & 0 & 0 \\ 0 & \beta_{i;t} & \pm\|b_{;i;q;t}\| e_1^T \\ 0 & \pm\|b_{;i;q;t}\| e_1 & \bar{B}_{;i;w;t} \end{pmatrix} \end{array}.$$

Next we apply a suitable elementary Gauss transform to complete the deflation of the eigenvector:

$$L_{;i;t}^{-1} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \rho_{i;t} e_1 & I \end{pmatrix} \end{array}, \qquad \rho_{i;t} = -\frac{\pm\|b_{;i;q}\|}{\beta_i}, \tag{25}$$

$$X_{;i+1;t} = \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} D_{;X;i;t} & 0 & 0 \\ 0 & \pm\|X_{i;i;t}\| & * \\ 0 & 0 & \bar{X}_{;i;w;t} \end{pmatrix} \end{array},$$

$$A_{;i+1;t} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;A;i;t} & 0 & 0 \\ 0 & \beta_{i;t}/\lambda_{i;i;t} & 0 \\ 0 & 0 & \bar{A}_{;i;w;t} - \frac{\|b_{;i;q;t}\|^2}{\lambda_{i;i;t}\beta_{i;t}} e_1 e_1^T \end{pmatrix} \end{array}, \tag{26}$$

$$B_{;i+1;t} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;B;i;t} & 0 & 0 \\ 0 & \beta_{i;t} & 0 \\ 0 & 0 & \bar{B}_{;i;w;t} - \frac{\|b_{;i;q;t}\|^2}{\beta_{i;t}} e_1 e_1^T \end{pmatrix} \end{array},$$

where

$$\left\| \frac{\|b_{;i;q;t}\|^2}{\lambda_{i;i;t}\beta_{i;t}} e_1 e_1^T \right\| \leq \left\| \frac{\|b_{;i;q;t}\|^2}{\beta_{i;t}} e_1 e_1^T \right\|, \tag{27}$$

since $|\lambda_{i;i;t}| \geq 1$.

Therefore, as long as $|\lambda_{i;i;t}| \geq 1$, the application of $L_{i;t}^{-1}$ to $A_{;i;w;t}$ is numerically stable (actually, backward stable with respect to A). This will follow from the more general analysis to be carried out below. We now analyze the case when $|\lambda_{i;i;t}| < 1$.

In step 6(a)(iv)(A) we perturb $B_{;i;q;t}$ such that its ith column is parallel to that of $A_{;i;q;t}$. Hence we have that

$$B_{;i;p;t} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;B;i;t} & 0 & 0 \\ 0 & \alpha_{i;t}\lambda_{i;i;t} & \lambda_{i;i;t}a_{;i;q;t}^T \\ 0 & \lambda_{i;i;t}a_{;i;q;t} & \bar{B}_{;i;q;t} \end{pmatrix} \end{array}. \tag{28}$$

It follows from (21) that

$$\|B_{;i;q;t} - B_{;i;p;t}\| \leq \epsilon(|\lambda_{i;i;t}|\|\bar{A}_{;i;t}\| + \|\bar{B}_{;i;t}\|). \tag{29}$$

The remaining steps in 6(a)(iv) are similar to the steps in 6(a)(iii) and we have accordingly that

$$W_{;i;t} \equiv \begin{array}{c} \\ i \\ n-i \end{array} \begin{array}{c} \begin{array}{cc} i & n-i \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & \bar{W}_{;i;t} \end{pmatrix} \end{array}, \qquad \bar{W}_i a_{;i;q;t} = \pm\|a_{;i;q;t}\|e_i, \tag{30}$$

$$X_{;i;w;t} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} D_{;X;i;t} & 0 & 0 \\ 0 & \pm\|X_{i;i;t}\| & h_{;i;t}^T \\ 0 & 0 & \bar{X}_{;i;w;t} \end{pmatrix} \end{array},$$

$$B_{;i;w;1;t} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;B;i;t} & 0 & 0 \\ 0 & \lambda_{i;i;t}\alpha_{i;t} & \pm\|a_{;i;q;t}\|\lambda_{i;i;t}e_1^T \\ 0 & \pm\|a_{;i;q;t}\|\lambda_{i;i;t}e_1 & \bar{B}_{;i;w;t} \end{pmatrix} \end{array},$$

$$A_{;i;w;t} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;A;i;t} & 0 & 0 \\ 0 & \alpha_{i;t} & \pm\|a_{;i;q;t}\|e_1^T \\ 0 & \pm\|a_{;i;q;t}\|e_1 & \bar{A}_{;i;w;t} \end{pmatrix} \end{array}.$$

Next we apply a suitable elementary Gauss transform to complete the deflation of the eigenvector:

$$L_{;i;t}^{-1} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \rho_{i;t}e_1 & I \end{pmatrix} \end{array}, \qquad \rho_{i;t} = -\frac{\pm\|a_{;i;q}\|}{\alpha_i}, \tag{31}$$

$$X_{;i+1;t} = \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} D_{;X;i;t} & 0 & 0 \\ 0 & \pm\|X_{i;i;t}\| & * \\ 0 & 0 & \bar{X}_{;i;w;t} \end{pmatrix} \end{array},$$

$$B_{;i+1;t} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;B;i;t} & 0 & 0 \\ 0 & \lambda_{i;i;t}\alpha_{i;t} & 0 \\ 0 & 0 & \bar{B}_{;i;w;t} - \frac{\lambda_{i;i;t}\|a_{;i;q;t}\|^2}{\alpha_{i;t}}e_1e_1^T \end{pmatrix} \end{array},$$

$$A_{;i+1;t} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} \bar{D}_{;A;i;t} & 0 & 0 \\ 0 & \alpha_{i;t} & 0 \\ 0 & 0 & \bar{A}_{;i;w;t} - \frac{\|a_{;i;q;t}\|^2}{\alpha_{i;t}} e_1 e_1^T \end{pmatrix} \end{array}. \tag{32}$$

This completes the proof by induction of (10) and (11). We are now ready to determine the global error of Algorithm II.

We first consider the case when the eigenvalues are larger than one in magnitude. Then the only error which affects the accuracy of the final answer is when we perturb $A_{;i;q;t}$ to get $A_{;i;p;t}$. Let

$$E_{;i;A;t} \equiv A_{;i;p;t} - A_{;i;q;t}, \tag{33}$$

where, by (23),

$$\|E_{;i;A;t}\| \leq \epsilon(\|\bar{A}_{;i;t}\| + \|\bar{B}_{;i;t}\|). \tag{34}$$

We will show that this error can be written as a small backward error in $A_{;i-1;p;t}$ or $A_{;i-1;t}$, depending on whether $|\lambda_{i-1;i-1;t}|$ was strictly smaller than one or not. Let

$$A_{;i-1;s;t} \equiv \begin{cases} A_{;i-1;t} & \text{if } |\lambda_{i-1;i-1;t}| < 1, \\ A_{;i-1;p;t} & \text{otherwise,} \end{cases}$$

and extend the definition of $E_{;i;A;t}$ by $E_{;i;A;t} \equiv A_{;i;s;t} - A_{;i;q;t}$. From an examination of Algorithm II it follows that

$$A_{;i;q;t} = L_{;i-1;t}^{-1} W_{;i-1;t} Q_{;i-1;t} A_{;i-1;s;t} Q_{;i-1;t}^T W_{;i-1;t}^T L_{;i-1;t}^{-T}.$$

Substituting in (33) we get

$$A_{;i;p;t} = L_{;i-1;t}^{-1} W_{;i-1;t} Q_{;i-1;t} (A_{;i-1;s;t} + E_{;i;A;b;t}) Q_{;i-1;t}^T W_{;i-1;t}^T L_{;i-1;t}^{-T}, \tag{35}$$

where

$$E_{;i;A;b;t} = Q_{;i-1;t}^T W_{;i-1;t}^T L_{;i-1;t} E_{;i;A;t} L_{;i-1;t}^T W_{;i-1;t} Q_{;i-1;t}.$$

From the structure equations (19), (22), and (25), we can conclude that

$$L_{;i-1;t} E_{;i;A;t} L_{;i-1;t}^T = E_{;i;A;t}.$$

Therefore,

$$\|E_{;i;A;b;t}\| = \|E_{;i;A;t}\|. \tag{36}$$

Furthermore, observe that $E_{;i;A;b;t}$, like $E_{;i;A;t}$, is nonzero only in the lower-right $(n-i+1) \times (n-i+1)$ block. Therefore, it follows from the structure equations (18), (24), and (25) that the matrix

$$Q_{;i-2;t}^T W_{;i-2;t}^T L_{;i-2;t} E_{;i;A;b;t} L_{;i-2;t}^T W_{;i-2;t} Q_{;i-2;t}$$

is nonzero only in the lower-right $(n-i+1) \times (n-i+1)$ block, and that

$$L_{;i-2;t} E_{;i;A;b;t} L_{;i-2;t}^T = E_{;i;A;b;t}.$$

Since this can be extended by induction, we can conclude using (35) and (36) that

$$D_{;A;t} = A_{;n;p;t} = C_{;t}^{-1}(A + E_{;A;t})C_{;t}^{-T},$$

where

$$\|E_{;A;t}\| \leq \sum_{i=1}^{n} \mu_i \|E_{;i;A;t}\| \leq \epsilon \left(\sum_{i=1}^{n} \mu_i \|\bar{A}_{;i;t}\| + \sum_{i=1}^{n} \mu_i \|\bar{B}_{;i;t}\| \right).$$

The last inequality follows from (34), and μ_i is defined as follows:

$$\mu_i = \begin{cases} 1 & \text{if } |\lambda_{i;i;t}| \geq 1, \\ 0 & \text{otherwise.} \end{cases}$$

We can carry out a similar analysis when $|\lambda_{i;i;t}| < 1$. Now the only error which affects the accuracy of the final answer is

$$E_{;i;B;t} \equiv B_{;i;p;t} - B_{;i;q;t}, \tag{37}$$

where, by (29),

$$\|E_{;i;B;t}\| \leq \epsilon(|\lambda_{i;i;t}| \ \|\bar{A}_{;i;t}\| + \|\bar{B}_{;i;t}\|). \tag{38}$$

We will show that this error can be written as a small backward error in $B_{;i-1;p;t}$ or $B_{;i-1;t}$, depending on whether $|\lambda_{i-1;i-1;t}|$ was strictly smaller than one or not. Let

$$B_{;i-1;s;t} \equiv \begin{cases} B_{;i-1;t} & \text{if } |\lambda_{i-1;i-1;t}| \geq 1, \\ B_{;i-1;p;t} & \text{otherwise,} \end{cases}$$

and extend the definition of $E_{;i;B;t}$ by $E_{;i;B;t} \equiv B_{;i;s;t} - B_{;i;q;t}$. From an examination of Algorithm II it follows that

$$B_{;i;q;t} = L_{;i-1;t}^{-1} W_{;i-1;t} Q_{;i-1;t} B_{;i-1;s;t} Q_{;i-1;t}^{T} W_{;i-1;t}^{T} L_{;i-1;t}^{-T}.$$

Substituting in (37) we get

$$B_{;i;p;t} = L_{;i-1;t}^{-1} W_{;i-1;t} Q_{;i-1;t} (B_{;i-1;s;t} + E_{;i;B;b;t}) Q_{;i-1;t}^{T} W_{;i-1;t}^{T} L_{;i-1;t}^{-T}, \tag{39}$$

where

$$E_{;i;B;b;t} = Q_{;i-1;t}^{T} W_{;i-1;t}^{T} L_{;i-1;t} E_{;i;B;t} L_{;i-1;t}^{T} W_{;i-1;t} Q_{;i-1;t}.$$

From the structure equations (20), (28), and (31), we can conclude that

$$L_{;i-1;t} E_{;i;B;t} L_{;i-1;t}^{T} = E_{;i;B;t}.$$

Therefore,

$$\|E_{;i;B;b;t}\| = \|E_{;i;B;t}\|. \tag{40}$$

Furthermore, observe that $E_{;i;B;b;t}$, like $E_{;i;B;t}$, is nonzero only in the lower-right $(n-i+1) \times (n-i+1)$ block. Therefore it follows from the structure equations (18), (30), and (31) that the matrix

$$Q_{;i-2;t}^{T} W_{;i-2;t}^{T} L_{;i-2;t} E_{;i;B;b;t} L_{;i-2;t}^{T} W_{;i-2;t} Q_{;i-2;t}$$

is nonzero only in the lower-right $(n-i+1)\times(n-i+1)$ block, and that

$$L_{;i-2;t}E_{;i;B;b;t}L_{;i-2;t}^T = E_{;i;B;b;t}.$$

Since this can be extended by induction, we can conclude using (39) and (40) that

$$D_{;B;t} = B_{;n;p;t} = C_{;t}^{-1}(B + E_{;B;t})C_{;t}^{-T},$$

where

$$\begin{aligned}\|E_{;B;t}\| &\leq \sum_{i=1}^{n}(1-\mu_i)\|E_{;i;B;t}\| \\ &\leq \epsilon\left(\sum_{i=1}^{n}(1-\mu_i)|\lambda_{i;i;t}|\ \|\bar{A}_{;i;t}\| + \sum_{i=1}^{n}(1-\mu_i)\|\bar{B}_{;i;t}\|\right). \qquad (41)\end{aligned}$$

The last inequality follows from (38).

Therefore, the error analysis boils down to obtaining good bounds on $\|\bar{A}_{;i;t}\|$ and $\|\bar{B}_{;i;t}\|$. We proceed to do that now.

We shall assume that

$$\|B^{-1}\|^{-1} > \|E_{;B;t}\|, \qquad (42)$$

so that $B_{;i;t}$ is always symmetric positive definite. (We shall give more practical conditions later.) Therefore, $\bar{B}_{;i;t}$ is the Schur complement of a symmetric positive-definite matrix and it follows that

$$\sum_{i=1}^{n}\|\bar{B}_{;i;t}\| \leq n\|B\| + O(\epsilon). \qquad (43)$$

Now consider

$$\sum_{i=1}^{n}(1-\mu_i)|\lambda_{i;i;t}|\ \|\bar{A}_{;i;t}\|.$$

From (26) and (32) we obtain

$$\bar{A}_{;i+1;t} = \bar{A}_{;i;w;t} - (1-\mu_i)\frac{\|a_{;i;q;t}\|^2}{\alpha_{i;t}}e_1e_1^T - \mu_i\frac{\|b_{;i;q;t}\|^2}{\lambda_{i;i;t}\beta_{i;t}}e_1e_1^T. \qquad (44)$$

By the assumption (42), $B_{;i;w;t}$ and $B_{;i;w;1;t}$ are symmetric positive definite. Hence we have that

$$\mu_i\left\|\frac{\|b_{;i;q;t}\|^2}{\lambda_{i;i;t}\beta_{i;t}}e_1e_1^T\right\| \leq \mu_i\|B_{;i;w;t}\| = \mu_i\|B\| + O(\epsilon) \qquad (45)$$

and

$$(1-\mu_i)\left\|\lambda_{i;i;t}\frac{\|a_{;i;q;t}\|^2}{\alpha_{i;t}}e_1e_1^T\right\| \leq (1-\mu_i)\|B_{;i;w;1;t}\| = (1-\mu_i)\|B\| + O(\epsilon). \qquad (46)$$

From (44), (45), and (46), we get

$$\begin{aligned}|\lambda_{i+1;i+1;t}|\ \|\bar{A}_{;i+1;t}\| &\leq \left|\frac{\lambda_{i+1;i+1;t}}{\lambda_{i;i;t}}\right|\ |\lambda_{i;i;t}|\ \|\bar{A}_{;i;t}\| \\ &\quad + \left((1-\mu_i)\left|\frac{\lambda_{i+1;i+1;t}}{\lambda_{i;i;t}}\right| + \mu_i\right)\|B\| + O(\epsilon). \qquad (47)\end{aligned}$$

Let k be the smallest integer such that $|\lambda_{k;k;t}| < 1$. Then it follows from (44) and (45) that

$$\|\bar{A}_{;i;t}\| \leq \|A\| + i\|B\|, \qquad i < k, \tag{48}$$

$$|\lambda_{k;k;t}|\ \|\bar{A}_{;k;t}\| \leq \|A\| + k\|B\|. \tag{49}$$

Now let

$$\tau_i \equiv \max\left(1, \left|\frac{\lambda_{i+1;i+1;t}}{\lambda_{i;i;t}}\right|\right).$$

Then we get from (47) and (49) that

$$|\lambda_{i+1;i+1;t}|\ \|\bar{A}_{;i+1;t}\| \leq (\|A\| + 2i\ \|B\|)\ \Pi_{j=k}^{i}\tau_j + O(\epsilon), \quad i \geq k. \tag{50}$$

This inequality is very loose.

Note that we cannot expect $\tau_i = 1$, because Algorithm II does *not* compute the exact eigenvalues of the pencil. From (58) it follows that

$$\tau_i \leq 1 + c_1\epsilon\kappa(\bar{B}_{;i;t}) + c_2\epsilon|\lambda_{i;i;t}|\ \|\bar{A}_{;i;t}\|\ \|\bar{B}_{;i;t}^{-1}\| + O(\epsilon^2),$$

where c_1 and c_2 are constants. Using (47), (49), and the fact that $\|A\| = \|B\|$, we obtain

$$\tau_i \leq 1 + c_3 i\epsilon\kappa(B) + O(\epsilon^2) \qquad \text{for } i \geq k,$$

where c_3 is a constant. Henceforth we make the assumption

$$c_3 n^2 \kappa(B)\epsilon < 1, \tag{51}$$

which is stronger than the previous assumption (42). Under this assumption it follows that

$$\Pi_{j=k}^{i}\tau_j \leq 2 + O(\epsilon^2).$$

Therefore, we can conclude from (50) that

$$\sum_{i=1}^{n}(1-\mu_i)|\lambda_{i;i;t}|\ \|\bar{A}_{;i;t}\| \leq 2n(\|A\| + n\|B\|) + O(\epsilon). \tag{52}$$

Substituting (52) and (43) in (41) we obtain

$$\|E_{;B;t}\| = \|C_{;t}D_{;B;t}C_{;t}^T - B\| \leq 2n\epsilon(\|A\| + n\|B\|) + O(\epsilon^2). \tag{53}$$

In other words the factorization $C_{;t}D_{;B;t}C_{;t}^T$ is backward stable (to first order) with respect to B, which is the first claim (8) of step 7 of Algorithm II. The main difficulty in proving this claim is that the perturbation in B depends upon the norms of the Schur complements of A, which can theoretically grow exponentially. The surprise is that the decomposition of B is backward stable *in spite of* this possibility.

We now turn our attention to proving the second claim (9), namely, that each eigenpair has a small residual with respect to A and B.

Let $C_{;i;t}$ denote the partially accumulated transform

$$C_{;i;t} \equiv Q_{;1;t}^T W_{;1;t}^T L_{;1;t} \cdots Q_{;i;t}^T W_{;i;t}^T L_{;i;t}^T.$$

Then note that the ith eigenvector is also given by

$$C_{;t}^{-1} e_i = C_{;i;t}^{-1} e_i.$$

This follows from (18), (24), and (25). Therefore, it is enough to prove that

$$\|(\lambda_{i;i;t} A - B) C_{;i;t}^{-T} e_i\| \leq 5n^2 \epsilon \, \|C_{;i;t}^{-T} e_i\| \, (|\lambda_{i;i;t}| \, \|A\| + \|B\|) + O(\epsilon^2)$$

in order to establish our claim. From (10) and (11), and the arguments leading to (36) and (40), it follows that

$$\|A - C_{;i;t} A_{;i+1;t} C_{;i;t}^T\| \leq \sum_{l=1}^{i} \|E_{;l;A;t}\|,$$

$$\|B - C_{;i;t} B_{;i+1;t} C_{;i;t}^T\| \leq \sum_{l=1}^{i} \|E_{;l;B;t}\|.$$

Therefore,

$$\begin{aligned} \|(\lambda_{i;i;t} A - B) C_{;i;t}^{-T} e_i\| \leq{}& \|(\lambda_{i;i;t} D_{i,i;A;t} - D_{i,i;B;t}) C_{;i;t} e_i\| \\ &+ \|C_{;i;t}^{-T} e_i\| \sum_{l=1}^{i} (|\lambda_{i;i;t}| \, \|E_{;l;A;t}\| + \|E_{;l;B;t}\|). \end{aligned}$$

Since

$$\lambda_{i;i;t} D_{i,i;A;t} = D_{i,i;B;t},$$

we have

$$\|(\lambda_{i;i;t} A - B) C_{;i;t}^{-T} e_i\| \leq \|C_{;i;t}^{-T} e_i\| \sum_{l=1}^{i} (|\lambda_{i;i;t}| \, \|E_{;l;A;t}\| + \|E_{;l;B;t}\|).$$

Note that $\|E_{;l;A;t}\|$ is multiplied by $|\lambda_{i;i;t}|$. This is crucial in the analysis which follows. From inequalities (41) and (53) it follows that

$$\sum_{l=1}^{i} \|E_{;l;B;t}\| \leq 2n\epsilon(\|A\| + n\|B\|) + O(\epsilon^2).$$

Now using our previous definition that k is the smallest integer such that $|\lambda_{k;k;t}| < 1$, we observe that if $i < k$, then from (34), (43), and (48) it follows that

$$|\lambda_{i;i;t}| \sum_{l=1}^{i} \|E_{;l;A;t}\| \leq |\lambda_{i;i;t}| \, 2n\epsilon(\|A\| + n\|B\|) + O(\epsilon^2), \qquad i < k.$$

When $i \geq k$, it follows from (49) that

$$|\lambda_{i;i;t}| \, \|\bar{A}_{;l;t}\| = \frac{|\lambda_{i;i;t}|}{|\lambda_{l;l;t}|} |\lambda_{l;l;t}| \, \|\bar{A}_{;l;t}\| \leq 2(\|A\| + 2l\|B\|) + O(\epsilon^2), \qquad l \leq i.$$

Combining this with (34) we obtain

$$\sum_{l=1}^{i} |\lambda_{i;i;t}| \, \|E_{;l;A;t}\| \leq 2n\epsilon(\|A\| + n\|B\|) + O(\epsilon^2), \qquad i \geq k.$$

Therefore, we can conclude that

$$\|(\lambda_{i;i;t}A - B)C_{;i;t}^{-T}e_i\| \leq \|C_{;i;t}^{-T}e_i\|(2 + |\lambda_{i;i;t}|)2n\epsilon(\|A\| + n\|B\|) + O(\epsilon^2).$$

Since $\|A\| = \|B\|$, this can be rewritten as

$$\|(\lambda_{i;i;t}A - B)C_{;i;t}^{-T}e_i\| \leq 5n^2\epsilon \, \|C_{;i;t}^{-T}e_i\| \, (|\lambda_{i;i;t}| \, \|A\| + \|B\|) + O(\epsilon^2),$$

which establishes our claim that each eigenpair has a small residual with respect to A and B. This is true *in spite of* the fact that the factorization of A is potentially unstable theoretically!

In fact using an analysis similar to that used to establish (53) we can show that

$$\|E_{;A;t}\| \leq p(n)\gamma(\|A\| + \|B\|)\epsilon,$$

where $p(n)$ is a low-order polynomial in n, and γ is the "growth factor" defined to be

$$\gamma \equiv \max_i \frac{\|\bar{A}_{;i;t}\|}{\|A\|}.$$

From (7) it follows that γ can grow at most exponentially in n, but in numerous numerical experiments (see section 6) we have never observed it. This is similar to the situation in Gaussian elimination with partial pivoting where the growth factor can potentially grow exponentially but is rarely observed [4]. Nevertheless, our algorithm will compute all eigenpairs reliably no matter how large or small γ happens to be.

We have now established that Algorithm II computes *each* eigenpair to backward accuracy (see (9)), and that it also computes *all* the eigenvectors, in the sense that they diagonalize B to backward accuracy (see (8)). Furthermore, by establishing inequality (see (12)), we have proved that Algorithm II is an $O(mn^3)$ algorithm, where m is the number of times step 6(b) is executed ($m < n$).

In extensive numerical experiments with Algorithm II the largest m we have observed is 7 for a pencil of size 500 by 500. Usually m is smaller than 4. Therefore, we conjecture that m is $O(\log(1/\epsilon))$. The reasoning for the conjecture is as follows.

We examine how many of the eigenvectors computed in step 6(b) will pass the residual test in step 6(a).

The analysis is an extension of the one used to prove inequality (14). We will continue to use the same notation. Immediately after step 6(b) we have (15), which we repeat here for convenience:

$$\bar{B}_{;i;t}\bar{X}_{j;i;t} + \bar{U}_{;i;t}\sqrt{\bar{\Sigma}_{;i;l;t}}\bar{E}_{;i;t}\bar{V}_{j;i;t} = \lambda_{j;i;l;t}\bar{A}_{;i;t}\bar{X}_{j;i;t}, \quad j = 1, \ldots, n-i+1. \tag{15}$$

Furthermore, from (13) we have that

$$\sqrt{\bar{\Sigma}_{;i;l;t}}\bar{U}_{;i;t}^T\bar{A}_{;i;t}^{-1}\bar{U}_{;i;t}\sqrt{\bar{\Sigma}_{;i;l;t}} - \bar{V}_{;i;t}\bar{\Lambda}_{;i;l;t}\bar{V}_{;i;t}^T = \bar{E}_{;i;t},$$

where

$$\|\bar{E}_{;i;t}\| \leq \epsilon\|\bar{\Lambda}_{;i;l;t}\|.$$

Therefore,

$$\begin{aligned}|\lambda_{j;i;l;t}| - \epsilon\|\bar{\Lambda}_{;i;l;t}\| &\leq \left\|\bar{U}_{;i;t}\sqrt{\bar{\Sigma}_{;i;l;t}}\right\| \, \left\|\bar{A}_{;i;t}^{-1}\bar{U}_{;i;t}\sqrt{\bar{\Sigma}_{;i;l;t}}\bar{V}_{j;i;t}\right\| \\ &= \left\|\bar{U}_{;i;t}\sqrt{\bar{\Sigma}_{;i;l;t}}\right\| \, \left\|\bar{X}_{j;i;t}\right\|.\end{aligned}$$

Hence

$$\frac{\|\bar{U}_{;i;t}\sqrt{\bar{\Sigma}_{;i;l;t}}\bar{E}_{;i;t}\bar{V}_{j;i;t}\|}{\|\bar{X}_{j;i;t}\|} \leq \frac{\epsilon}{\frac{|\lambda_{j;i;l;t}|}{\|\bar{\Lambda}_{;i;l;t}\|} - \epsilon}\|\bar{\Sigma}_{;i;l;t}\|. \tag{54}$$

From the variational characterization of eigenvalues we have

$$\|\bar{\Lambda}_{;i;l;t}\|(1+\epsilon) \geq \frac{\|\bar{B}_{;i;t}\|}{\|\bar{A}_{;i;t}\|}.$$

Therefore,

$$\begin{aligned}|\lambda_{j;i;l;t}|\|\bar{A}_{;i;t}\| + \|\bar{B}_{;i;t}\| &= \frac{|\lambda_{j;i;l;t}|}{|\lambda_{1;i;l;t}|}|\lambda_{1;i;l;t}|\|\bar{A}_{;i;t}\| + \|\bar{B}_{;i;t}\| \\ &\geq \frac{|\lambda_{j;i;l;t}|}{|\lambda_{1;i;l;t}|}\frac{\|\bar{B}_{;i;t}\|}{1+\epsilon} + \|\bar{B}_{;i;t}\| \\ &= \|\bar{B}_{;i;t}\|\frac{\frac{|\lambda_{j;i;l;t}|}{|\lambda_{1;i;l;t}|} + 1 + \epsilon}{1+\epsilon}. \end{aligned} \tag{55}$$

Now, if

$$\frac{|\lambda_{j;i;l;t}|}{|\lambda_{1;i;l;t}|} > \frac{\sqrt{5}-1}{2} + 3\epsilon, \tag{56}$$

then

$$\frac{1}{\frac{|\lambda_{j;i;l;t}|}{\|\bar{\Lambda}_{;i;l;t}\|} - \epsilon} < \frac{\frac{|\lambda_{j;i;l;t}|}{|\lambda_{1;i;l;t}|} + 1 + \epsilon}{1+\epsilon}.$$

Therefore, from (15), (54), and (55), we can conclude that

$$\|(\lambda_{j;i;t}\bar{A}_{;i;t} - \bar{B}_{;i;t})\bar{X}_{j;i;t}\| \leq \epsilon\|\bar{X}_{;i;t}\|(|\lambda_{j;i;t}|\ \|\bar{A}_{;i;t}\| + \|\bar{B}_{;i;t}\|)$$

for all j between 1 and $n-i+1$ such that inequality (56) is satisfied.

In other words, the eigenvectors corresponding to the eigenvalues in a *significant* interval around the largest eigenvalue will be sufficiently accurate to pass the residual test. The algorithm proceeds by deflating the eigenvector corresponding to the largest eigenvalue in magnitude, $\lambda_{1;i;t}$. The important question now is how many of the originally accurate eigenvectors will continue to pass the residual test after the deflation. In numerical experiments most of these eigenvectors continue to pass the residual test after the deflation. Our best explanation of this phenomenon is that the eigenvectors seem to be ordered in rank-revealed form and that the *total* loss of accuracy due to the nonorthogonal transforms is $O(\kappa(B)\epsilon)$. This explains our conjecture. Doubtless, more extensive investigations are needed and they are being carried out.

5. Implementation issues. Algorithm II was structured to make the error analysis easy. Here we present some necessary details for a practical implementation.

Singular A. So far, throughout our analysis, we had assumed that $\bar{A}_{;i;t}$ was nonsingular. This is hard to maintain, since we have no real control on the growth of the condition number of $\bar{A}_{;i;t}$. Furthermore, the additive perturbations can make some $\bar{A}_{;i;t}$ singular. Therefore, we need to explicitly deal with singular $\bar{A}_{;i;t}$.

In step 6(b) we explicitly check if the condition number of $\bar{A}_{;i;t}$ is bigger than a small constant times the reciprocal of the machine precision. If it is, we compute all the eigenvectors of $\bar{A}_{;i;t}$ corresponding to its small eigenvalues (of the size of the machine precision). Compute the Householder transform which maps the eigenvector corresponding to the smallest eigenvalue in magnitude to e_1. Apply this to $\bar{A}_{;i;t}$ and $\bar{B}_{;i;t}$. The first column and row of $\bar{A}_{;i;t}$ will be numerically zero. They should be explicitly set to zero. Now by elementary Gauss transforms the first column and row of $\bar{B}_{;i;t}$ are eliminated. This will have no effect on $\bar{A}_{;i;t}$. This process is continued until all the eigenvectors of $\bar{A}_{;i;t}$ corresponding to the small eigenvalues (in magnitude) have been deflated. Note that this procedure has no adverse effect on the accuracy and speed of Algorithm II.

Residual test. The residual test in step 6(a) is too strong and should be weakened as follows:

$$\|(\lambda_{i;i;t}A_{;i;t} - B_{;i;t})X_{i;i;t}\| \leq \epsilon\|X_{i;i;t}\|(|\lambda_{i;i;t}| \max_{j\leq i}\|\bar{A}_{;j;t}\| + \max_{j\leq i}\|\bar{B}_{;j;t}\|).$$

This is to account for the fact that the Schur complements of A and B might decrease in norm during the deflation process.

Well-conditioned submatrices. If either $\bar{A}_{;i;t}$ or $\bar{B}_{;i;t}$ happen to be well-conditioned, then it improves performance to directly compute the eigenvalues and eigenvectors of either $\bar{B}_{;i;t}\bar{A}^{-1}_{;i;t}$ or $\bar{A}_{;i;t}\bar{B}^{-1}_{;i;t}$, respectively (in symmetric form, of course), and avoid deflating them.

QR factorization of $C_{;t}$. The matrix $C_{;t}$ can be stored in factorized form as in step 7 of Algorithm II, or it can be stored in QR-factored form, which might be more convenient for the user. To do this we observe that

$$C_{;t} = Q^T_{;1;t}W^T_{;1;t}\cdots Q^T_{;n;t}W^T_{;n;t}\tilde{L}_{;1;t}\cdots\tilde{L}_{;n;t},$$

where $\tilde{L}_{;i;t}$ is still an elementary Gauss transform of the type

$$\tilde{L}_{;i;t} \equiv \begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & y_{;i;t} & I \end{pmatrix} \end{array},$$

and $y_{;i;t}$ can be obtained easily from $\rho_{i;t}$ and the transforms $Q_{;j;t}$ and $W_{;j;t}$ for $j > i$. Due to this, the QR factorization of $C_{;t}$ can be computed efficiently in $O(n^3)$ flops.

Numerical issues. So far we have tacitly assumed that we can compute the large eigenvalues and corresponding eigenvectors of $\sqrt{\Sigma_{;i;t}}U^T_{;i;t}A^{-1}_{;i;t}U_{;i;t}\sqrt{\Sigma_{;i;t}}$ in step 6(b)(ii) to sufficient accuracy. That task is nontrivial numerically, and in this section we detail the mechanism for carrying it out.

We first compute the eigendecomposition of $A_{;i;t} = Z_{;i}\Delta_{;i}Z^T_{;i}$. Then we compute the required product as

$$\sqrt{\Sigma_{;i;t}}\left((U^T_{;i;t}Z_{;i})\Delta_{;i}(Z^T_{;i}U_{;i;t})\right)\sqrt{\Sigma_{;i;t}}$$

in the order suggested by the parentheses. It is important to follow the suggested order of operations. When the expression $(U^T_{;i;t}Z_{;i})\Delta_{;i}(Z^T_{;i}U_{;i;t})$ is being evaluated it should be done in outer-product form, exploiting the fact that $\Delta_{;i}$ is diagonal. Then a round-off error analysis can be used to show that if $Y_{;i}$ denotes the computed product, then

$$Y_{;i} + E_{;Y;i} = \sqrt{\Sigma_{;i;t}}U^T_{;i;t}(A_{;i;t} + E_{;A;m;i})^{-1}U_{;i;t}\sqrt{\Sigma_{;i;t}},$$

where

$$\|E_{;A;m;i}\| \leq n^2\epsilon\|A_{;i;t}\|,$$
$$\|E_{;Y;i}\| \leq n^2\epsilon\|\sqrt{\Sigma_{;i;t}}\|\ \|A_{;i;t}^{-1}U_{;i;t}\sqrt{\Sigma_{;i;t}}\|.$$

Since $\|Y_{;i}\|$ can be as small as $\sigma_{\min}(\Sigma_{;i;t})/\|A_{;i;t}\|$, it follows that the computed $Y_{;i}$ can fail to be stable by approximately $\epsilon\sqrt{\kappa(\Sigma_{;i;t})}$ digits. Since we have assumed that $\kappa(B)\epsilon < 1$, it follows that $Y_{;i}$ is accurate to at least half the digits. Therefore, if the computed eigenvectors of $Y_{;i}$ corresponding to the largest eigenvalues turn out to be too inaccurate to pass the residual test, we can efficiently correct them by inverse iteration (see chapter 5 in [1]).

Round-off errors. The analysis of the round-off errors incurred in the algorithm is similar to the analysis of the eigenvalue decomposition truncation errors carried out for Algorithm II. We just combine the standard error analysis techniques of Gaussian elimination with the truncation error analysis of section 4. The final error bounds have larger polynomials in n.

6. Numerical experiments. We now describe some numerical experiments that were carried out to test the accuracy and efficiency of Algorithm II. The algorithm was implemented in Matlab [5] and run on a Sun SPARCstation 20. The machine precision was approximately 10^{-16} (denoted by ϵ_{mach}). The algorithm was tested on three classes of randomly generated pencils with different characteristics.

The results for the first class of test matrices are shown in Figure 1, those for the second class of test matrices in Figure 2, and those for the third class of test matrices in Figure 3. In each class the matrices of all sizes from 10 to 100 were tested. In all the figures the horizontal axis represents the matrix size.

Figures 1(a), 2(a), and 3(a) show $n\kappa(A)$ for the experimental run being reported, where n denotes the matrix size.

Figures 1(b), 2(b), and 3(b) similarly show $n\kappa(B)$.

Figures 1(c), 2(c), and 3(c) show the error in the factorization of A, which is defined as follows:

$$\frac{\|A - C_{;t}D_{A;t}C_{;t}^T\|}{\|A\|}\frac{\epsilon_{mach}}{n\epsilon}.$$

The reason for the normalization factor is that from inequality (8) we expect the error to be bounded by a quadratic polynomial in n times ϵ. Since the bounds in error analysis tend to be conservative we chose to normalize by a linear polynomial instead.

Figures 1(d), 2(d), and 3(d) show the error in the factorization of B:

$$\frac{\|B - C_{;t}D_{B;t}C_{;t}^T\|}{\|B\|}\frac{\epsilon_{mach}}{n\epsilon}.$$

The plots in Figures 1(e), 2(e), and 3(e) display the accuracy of the computed eigenvalues measured by the expression

$$\max_i \frac{\sigma_{\min}(D_{i,i;B;t}A - D_{i,i;A;t}B)}{n\sigma_{\max}(D_{i,i;B;t}A - D_{i,i;A;t}B)}.$$

This expression measures the accuracy of the computed eigenvalues independently of the computed eigenvectors.

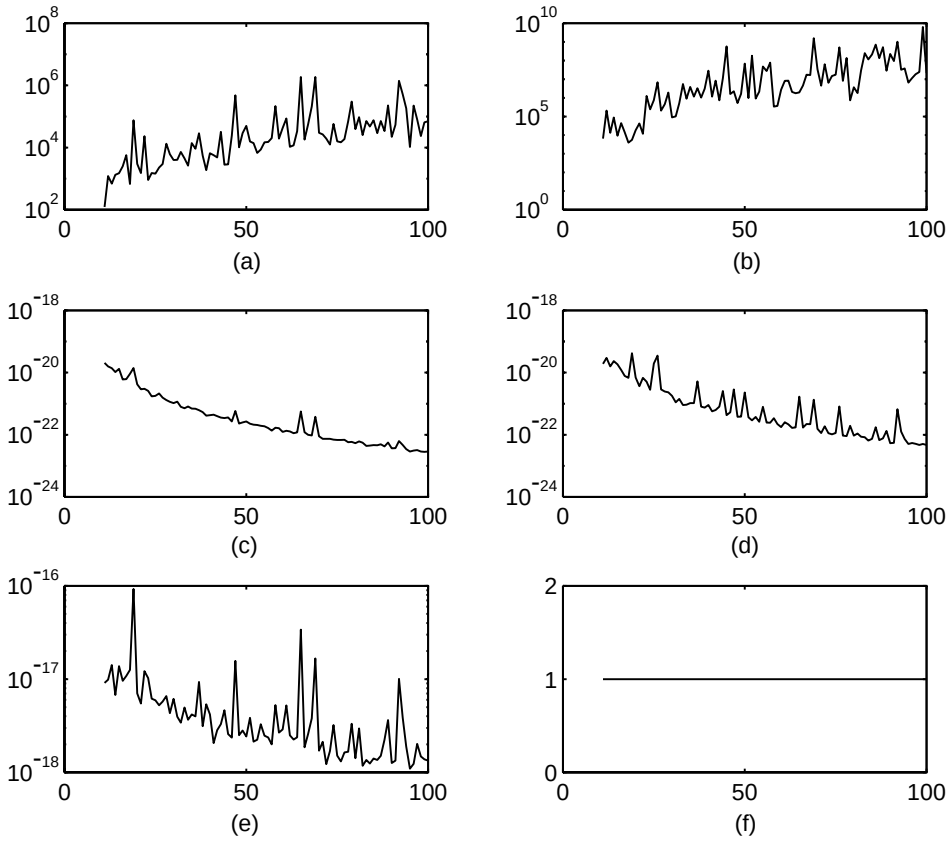

FIG. 1. *First test class.*

Finally, Figures 1(f), 2(f), and 3(f) display the number of times step 6(b) is executed. This gives an indication of the efficiency of the algorithm since the cost of Algorithm II is $O(mn^3)$, where m is the number of times step 6(b) is executed. We remind the reader that in step 6(b) we compute eigenvalue decompositions of dense matrices, and it can cost $O(n^3)$ flops. Hence it essentially determines how efficient the algorithm will be. Of course, we want to ensure that step 6(b) is executed as few times as possible.

For the experiments presented here we chose ϵ to be equal to $20n^{1.5}\epsilon_{mach}$. This was based on an estimate of the average accuracy of the solver for the symmetric eigenvalue problem. Choosing a smaller ϵ can make the algorithm inefficient with little gain in accuracy.

The first class of test matrices is generated as follows. To create the matrix A of order n, we generate a random (normal distribution) $n \times n$ matrix, take its symmetric part, and normalize it by its 1-norm. The matrix tends to have a moderately high condition number, as can be seen from Figure 1(a). To generate the matrix B of order n, we take a random (normal distribution) matrix of order n, multiply it by its transpose (to make it symmetric positive definite), and then normalize it by its 1-norm. The matrix tends to be more ill-conditioned than A, as shown in Figure 1(b). As can be seen from Figure 1(c) the computed eigenvectors diagonalize A in each instance to full backward accuracy. That is, there is no evidence of a large growth factor.

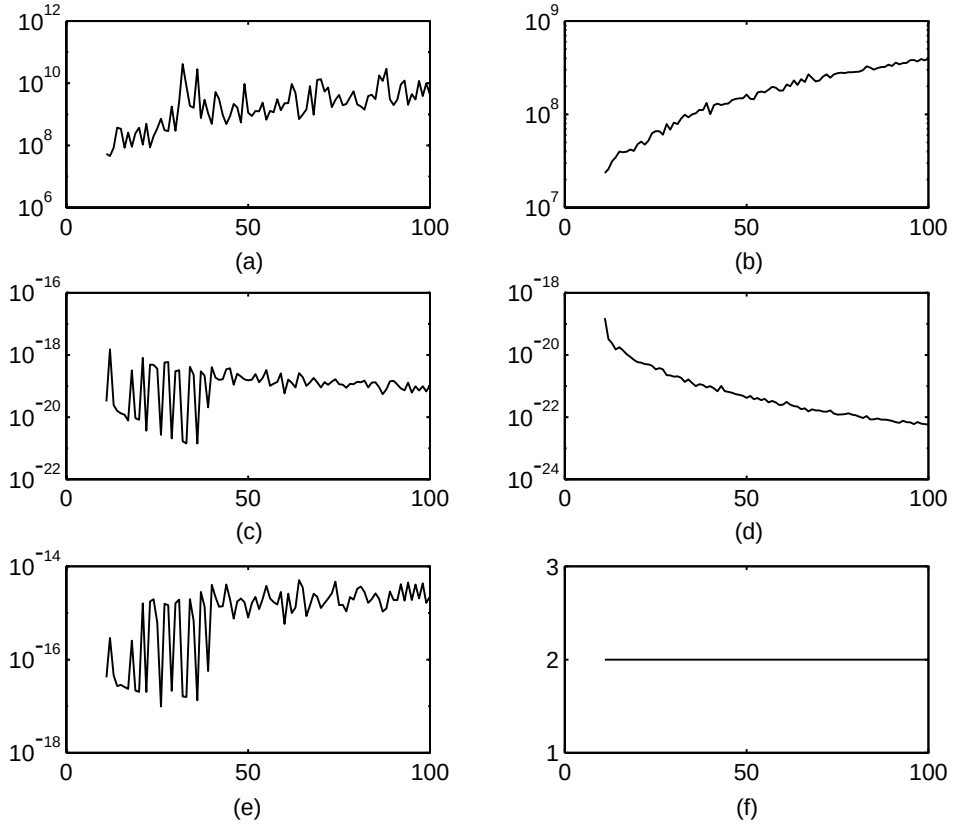

FIG. 2. *Second test class.*

As expected from the error analysis, Figure 1(d) demonstrates that the computed eigenvectors diagonalize B to full backward accuracy. From Figure 1(e) we see that the eigenvalues are computed with better accuracy than the error analysis indicates. Finally, Figure 1(f) shows that the algorithm is very efficient on this test class, as we never require more than one execution of step 6(b).

The second class of test matrices was generated as follows. Let $c_b \equiv 10^{-8}n$, where n denotes the order of the matrix. To create A, we first generated a random (normal distribution) matrix, took its symmetric part, and computed its eigenvalues, $\lambda_1 \leq \cdots \leq \lambda_n$. We then added $-\lambda_{[n/2]} + c_b\lambda_n$ times the identity matrix to the symmetric part, normalized the result by its 1-norm, and obtained A. The general result is a symmetric indefinite matrix with a rather large condition number, as seen in Figure 2(a). To create B, we generate a random (normal distribution) matrix of order n, extract its symmetric part, compute its eigenvalues, $\mu_1 \leq \cdots \leq \mu_n$, and add $|\mu_1| + c_b \max(\mu_1, \mu_n)$ times the identity to the symmetric part. This almost always results in a symmetric positive-definite matrix, with a moderately large condition number, as can be seen in Figure 2(b). Note that in the run shown in Figure 2, the matrix B is generally better conditioned than A, though both in general have rather large condition numbers. Again Figure 2(c) indicates that the computed eigenvectors diagonalize A to backward stability in each case, which is much better than what the error analysis predicts. Figure 2(d) shows that the eigenvectors diagonalize B to

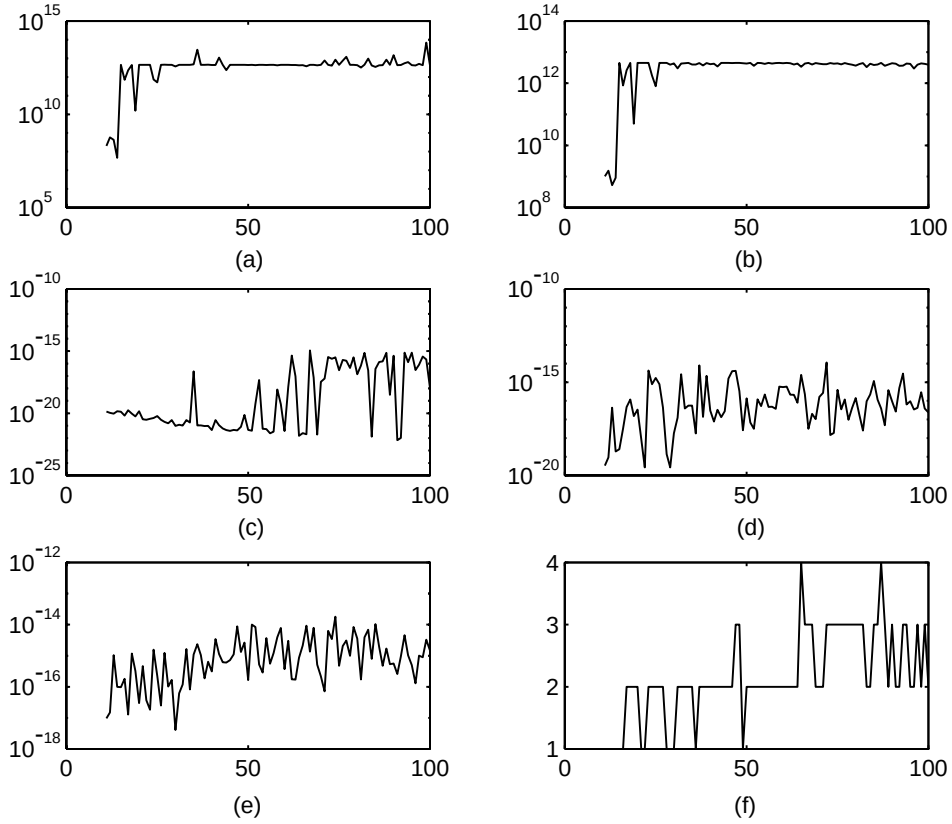

FIG. 3. *Third test class.*

full backward accuracy (as the error analysis predicts). Figure 2(e) shows that the individual eigenvalues are highly accurate. Figure 2(f) again shows the high efficiency of the algorithm. In this run we never execute step 6(b) more than two times. The main differences between the first test class and this one (the second) is that A is generally more ill-conditioned in the latter class and that B is made ill-conditioned by shifting (second class) rather than squaring (first class).

The third test class of matrices is deliberately chosen to try to make the algorithm perform badly. From the error analysis we identify the following features which such a test class must satisfy: First, the eigenvalues must not be too small or too large in magnitude; second, the eigenvalues must be highly clustered, preferably with eigenvalue gaps in a wide range; third, the eigenvalues must be highly ill-conditioned. We generate such a test class of matrices as follows. We first generate a random (uniform distribution) lower-triangular matrix, L. We then generate a diagonal matrix, D, whose ith element is a positive random number times $(-1.25)^i$. Then D is normalized by its 2-norm. The matrix A is taken to be equal to LDL^T. We now multiply the ith diagonal element of D by $1 + c_i$, where c_i is a random number uniformly distributed between -10^{-6} and 10^{-6}. The matrix B is taken to be equal to LDL^T, where we use the new D now. From the way we created A and B we can see that eigenvalues will be tightly clustered around $+1$ and -1. From Figures 3(a) and 3(b) we see that the matrices A and B are highly ill-conditioned, making the eigenvalues also highly

ill-conditioned. But from Figures 3(c) and 3(d) we see that the computed eigenvectors diagonalize A and B to full backward accuracy. From Figure 3(e) we see that eigenvalues are all accurate. From Figure 3(f) we see that now the algorithm executes step 6(b) more frequently than for the previous two test classes, but nevertheless the number of iterations never exceeds 4. This gives further evidence of the efficiency and robustness of the algorithm.

6.1. Some numerical examples. In this section we give some examples where conventional algorithms for the symmetric-definite generalized eigenvalue problem can give erroneous answers.

We first provide an example where Matlab's QZ algorithm computes eigenvalues with large imaginary parts. The algorithm described in this paper has no such problems and returns all the eigenvalues and eigenvectors to full backward accuracy. To proceed, we define the matrices A and B as follows:

$$A = \begin{pmatrix} -0.00116607575822 & -0.00039530283170 & -0.00076250811040 & 0.00112106013448 & -0.00065381674531 \\ -0.00039530283170 & -0.00013392875708 & -0.00025887606360 & 0.00038006880479 & -0.00022125054686 \\ -0.00076250811040 & -0.00025887606360 & -0.00049676987879 & 0.00073294540527 & -0.00042943322558 \\ 0.00112106013448 & 0.00038006880479 & 0.00073294540527 & -0.00107777377871 & 0.00062870745935 \\ -0.00065381674531 & -0.00022125054686 & -0.00042943322558 & 0.00062870745935 & -0.00036464347683 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 0.00116607575822 & 0.00039530283170 & 0.00076250811040 & -0.00112106013448 & 0.00065381674531 \\ 0.00039530283170 & 0.00013408869830 & 0.00025810855444 & -0.00038001602253 & 0.00022204073848 \\ 0.00076250811040 & 0.00025810855444 & 0.00050045292133 & -0.00073319869124 & 0.00042564133688 \\ -0.00112106013448 & -0.00038001602253 & -0.00073319869124 & 0.00107779119829 & -0.00062844668852 \\ 0.00065381674531 & 0.00022204073848 & 0.00042564133688 & -0.00062844668852 & 0.00036854742882 \end{pmatrix}.$$

When we computed the eigenvalues using "eig(A,B)" in Matlab, one of the eigenvalues it returned was $-0.96956258140331 + 0.00009155375727i$, which as one can see has a large imaginary part for a computation carried out with approximately 14 decimal digits of accuracy.

The next example is one where the conventional idea of converting $Ax = \lambda Bx$ into $G^{-1}AG^{-T}x = \lambda x$, where $B = GG^T$, fails to do well. Let

$$A = \begin{pmatrix} 0.21472417430628 & 0.17400616567923 & -0.02769100324675 & -0.13782812711118 & 0.34789671111569 \\ 0.17400616567923 & -0.79254771778534 & -0.06583322926616 & -0.21873063212490 & 0.52253219013903 \\ -0.02769100324675 & -0.06583322926616 & 0.30681582819807 & -0.85955850791110 & -0.39898863450352 \\ -0.13782812711118 & -0.21873063212490 & -0.85955850791110 & -0.09425601545807 & -0.38773515091776 \\ 0.34789671111569 & 0.52253219013903 & -0.39898863450352 & -0.38773515091776 & 0.36526373073906 \end{pmatrix}$$

and

$$B = \operatorname{diag} \begin{pmatrix} 1.00000000000000 \\ 0.00050000000000 \\ 0.00000025000000 \\ 0.00000000012500 \\ 0.00000000000006 \end{pmatrix}.$$

When this example is solved in Matlab using "eig($G^{-1}AG^{-T}$)," the largest normalized residual is approximately 10^{-9}, which is too big for a computation carried out in double-precision arithmetic for a 5×5 problem.

7. Perturbation of generalized eigenvalues. In this section we develop the necessary perturbation theory of generalized eigenvalues of symmetric-definite pencils.

Let

$$\begin{aligned} B &= GG^T, \\ G^T A^{-1} G &= V_1 \Lambda_1 V_1^T + E_1, \\ B + E_B &= GFFG^T, \quad F = F^T, \\ FG^T(A + E_A)^{-1} GF &= V_2 \Lambda_2 V_2^T + E_2, \end{aligned}$$

where

$$\begin{aligned}\|E_1\| &\le \epsilon\|\Lambda_1\|,\\ \|E_2\| &\le \epsilon\|\Lambda_2\|,\\ \|E_B\| &\le \epsilon\|B\|,\\ \|E_A\| &\le \epsilon\|A\|,\\ \epsilon\kappa(B) &< 1.\end{aligned}$$

Since $FF = I + G^{-1}E_BG^{-T}$, we have

$$G - GF = G(I - F) = -E_BG^{-T}(I + F)^{-1}.$$

Using the fact that F is symmetric positive definite, we have

$$\|GF - G\| \le \epsilon\frac{1}{2}\|B\|\,\|G^{-1}\| + O(\epsilon^2).$$

Therefore, we can expand to first order in ϵ to obtain

$$\begin{aligned}V_2\Lambda_2V_2^T + E_2 = {} & V_1\Lambda_1V_1^T + E_1 + (GF - G)^TA^{-1}G + G^TA^{-1}(GF - G)\\ & + G^TA^{-1}E_AA^{-1}G + O(\epsilon^2).\end{aligned} \tag{57}$$

Also, since

$$A^{-1}G = G^{-T}V_1\Lambda_1V_1^T + O(\epsilon),$$

we obtain from (57) that

$$\|\Lambda_2\|(1 - \epsilon) \le \|\Lambda_1\|(1 + \epsilon + \epsilon\kappa(B) + \epsilon\|\Lambda_1\|\,\|A\|\,\|B^{-1}\|) + O(\epsilon^2),$$

where $\kappa(B) = \|B\|\,\|B^{-1}\|$ is the condition number of B. Therefore,

$$\frac{\|\Lambda_2\|}{\|\Lambda_1\|} \le 1 + 3\epsilon\kappa(B) + \epsilon\|\Lambda_1\|\,\|A\|\,\|B^{-1}\| + O(\epsilon^2). \tag{58}$$

The important thing to observe here is that the first-order perturbation bound is not directly dependent on the condition number of A.

8. Conclusion. We have presented a new technique for the stable deflation of eigenpairs from the pencil $Ax = \lambda Bx$, where A is symmetric indefinite and B is symmetric positive definite. We have given an error analysis to prove the stability of the algorithm, as well as numerical evidence for its efficiency and robustness even for almost singular problems. Hence we have shown that numerically reliable and efficient software for the symmetric-definite generalized eigenvalue problem can be written without unnecessarily restricting the class of problems on which it can work. We also see that we do not need extra precision beyond what is available intrinsically in the hardware. In addition we presented a new eigenvalue perturbation bound.

Acknowledgments. We would like to thank Ming Gu for useful discussions and the referees for many constructive suggestions.

REFERENCES

[1] S. Chandrasekaran, *When is a Linear System Ill-Conditioned?*, Ph.D. thesis, Yale University, New Haven, CT, 1994.
[2] G. Golub and C. van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1996.
[3] A. Householder, *The Theory of Matrices in Numerical Analysis*, Dover, New York, 1964.
[4] W. Kahan, *Numerical linear algebra*, Canad. Math. Bull., 9 (1966), pp. 757–801.
[5] The MathWorks, Inc., *Matlab Reference Guide*, Natick, MA, 1992.
[6] B. Parlett, *The Symmetric Eigenvalue Problem*, Prentice–Hall, Englewood Cliffs, NJ, 1980.
[7] G. Stewart, *A bibliographical tour of the large, sparse generalized eigenvalue problem*, in Sparse Matrix Computations, J. Bunch and D. Rose, eds., Academic Press, New York, 1976, pp. 113–130.
[8] J. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.