An Algorithm for the Generalized Symmetric Tridiagonal Eigenvalue Problem

Kuiyuan Li
Department of Mathematics
and Statistics
The University of West Florida
Pansacola, FL 32514
kli@conch.senod.uwf.edu

Tien-Yien Li*
Department of Mathematics
Michigan State University
East Lansing, MI 48824
li@math.msu.edu

Zhonggang Zeng
Department of Mathematics
Northeastern Illinois University
Chicago, IL 60625
zeng@uxa.ecn.bgu.edu

Abstract

In this paper we present an algorithm, parallel in nature, for finding eigenvalues of a symmetric definite tridiagonal matrix pencil. Our algorithm employs the determinant evaluation, split-and-merge strategy and Laguerre's iteration. Numerical results on both single and multiprocessor computers are presented which show that our algorithm is reliable, efficient and accurate. It also enjoys the flexibility in evaluating partial spectrum.

Key words: eigenvalues, multiprocessors, matrix pencil.

Contents

*This research was supported in part by NSF under Grant CCR-9024840

1 Introduction

In this paper we consider the generalized eigenproblem

$$Tx = \lambda Sx \quad \lambda \in \boldsymbol{R} \text{ and } x \in \boldsymbol{R}^n \tag{1}$$

where both T and S are $n \times n$ symmetric tridiagonal matrices and one of them is positive definite. The pair (T, S) is called a symmetric tridiagonal definite pencil. For the rest of the paper, we shall assume the matrix S is positive definite.

Eigenproblem with such a special structure arises in many applications such as numerical solution of Sturm-Liouville and radial Schrödinger equations [2, 4, 5, 9, 22], finite element approximation for free longitudinal vibrations problem of non-uniform rod [19, 24], etc.

The generalized eigenproblem (1) can be reduced to a standard eigenvalue problem [8, 17, 25]

$$L^{-1}TL^{-T}(L^Tx) = \lambda(L^Tx) \tag{2}$$

where $S = LL^T$. Then eigenproblem (2) can be solved by many methods, such as the QR algorithm [17], the bisection/multisection method [15], the divide-conquer algorithm [6, 7], and homotopy like algorithms [12, 14]. However, this approach is less attractive because it can not take advantage of the tridiagonal form of T and S, and a full matrix $L^{-1}TL^{-T}$ is generated in the process. The accuracy of this method also depends on the condition of S since the inverse of L is required.

We propose an algorithm that computes the generalized eigenvalues of a symmetric definite tridiagonal pencil (T, S) through finding zeros of the polynomial equation

$$f(\lambda) \equiv \det\,[T - \lambda S] = 0 \tag{3}$$

where $f(\lambda)$, the characteristic polynomial of the pencil (T, S), and its derivatives can be evaluated by modified three term recurrences (§3) which virtually eliminate overflow and underflow problems. The equation (3) is solved by Laguerre's iteration [10, 14, 25] with starting points obtained by a split-merge process similar to Cuppen's divide-and-conquer method [6]. The resulting algorithm evaluates only eigenvalues and the inverse iteration is used when eigenvectors are also needed.

In [14], this method was fully developed for the special case when $S = I$, the standard eigenvalue problem, and numerical results on a substantial variety of matrices indicate that the algorithm is very efficient and accurate. The speed is sequentially competitive with the QR algorithm and faster than bisection/multisection [15] as well as Cuppen's divide-and-conquer algorithm [6, 7] by a considerable margin. Most importantly, our method is, in contrast to the highly serial QR algorithm, fully parallel and scalable [23]. For a general matrix S, magnitude of the eigenvalues of the pencil (T, S) can be arbitrarily large when S is nearly singular. It is, therefore, inappropriate to apply the algorithm developed in [14] directly to finding zeros of (3) when all eigenvalues of (T, S) are in demand. The modification of the algorithm for general S is given in §6. Moreover, the split-merge strategy used in [14] provides an interlacing property which gives an excellent set of starting points for the Laguerre iteration. This interlacing property needs to be reestablished for general S, which is described in §5. On the other hand, the backward and forward error analysis of our algorithm are shown in §4, the numerical results are presented in §7.

There are other approaches for this problem. In [3, 20], divide-and-conquer methods are proposed, and the bisection method is suggested in [25]. In more general cases, algorithms for banded pencil is also proposed in [11, 16, 18].

2 Laguerre's iteration

Since (T, S) is a symmetric definite pencil, all its eigenvalues, i.e., all zeros of the polynomial $f(\lambda)$ in (3), are real [17, p308]. Thus, Laguerre's iteration [25, pp443-445]

$$L_{\pm}(x) = x + \frac{n}{\left(-\frac{f'(x)}{f(x)}\right) \pm \sqrt{(n-1)\left[(n-1)\left(-\frac{f'(x)}{f(x)}\right)^2 - n\left(\frac{f''(x)}{f(x)}\right)\right]}} \tag{4}$$

appears perfectly suitable for this case. More specifically, let $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ be eigenvalues of the pencil (T, S), and let $\lambda_0 = -\infty$ and $\lambda_{n+1} = +\infty$. For any $x \in (\lambda_i, \lambda_{i+1})$, $i = 0, 1, \cdots, n$, one can use (4) to generate two sequences

$$x_+^{(k)} = L_+^k(x) \equiv \overbrace{L_+(L_+(\cdots L_+}^{k}(x)\cdots)) \quad \text{and} \quad x_-^{(k)} = L_-^k(x) \equiv \overbrace{L_-(L_-(\cdots L_-}^{k}(x)\cdots))$$

such that

$$\lambda_i \longleftarrow \cdots x_-^{(2)} < x_-^{(1)} < x < x_+^{(1)} < x_+^{(2)} \cdots \longrightarrow \lambda_{i+1}.$$

The convergence in each direction is cubic in a neighborhood of the corresponding zero as long as it is simple. If a zero, say λ_i, is an r-fold zero of $f(\lambda)$, the modified Laguerre iteration

$$L_{r\pm}(x) = x + \frac{n}{\left(-\frac{f'(x)}{f(x)}\right) \pm \sqrt{\frac{n-r}{r}\left[(n-1)\left(-\frac{f'(x)}{f(x)}\right)^2 - n\left(\frac{f''(x)}{f(x)}\right)\right]}} \tag{5}$$

can be used to generate sequences

$$x_{r+}^{(k)} = L_{r+}^k(x) \equiv \overbrace{L_{r+}(L_{r+}(\cdots L_{r+}}^{k}(x)\cdots))$$

and

$$x_{r-}^{(k)} = L_{r-}^k(x) \equiv \overbrace{L_{r-}(L_{r-}(\cdots L_{r-}}^{k}(x)\cdots)).$$

When $x_{r+}^{(0)} = x$ is less than λ_i, then

$$x_{r+}^{(1)} < x_{r+}^{(2)} < \cdots \longrightarrow \lambda_i,$$

and

$$x_{r-}^{(1)} > x_{r-}^{(2)} > \cdots \longrightarrow \lambda_i$$

if $x_{r-}^{(0)} = x$ is larger than λ_i. The convergence is still cubic in a neighborhood of λ_i. The estimate of the multiplicity r of λ_i can be obtained when the strategy of *split-merge* is employed (see §5 and §6).

The monotone convergence with ultimate cubic convergence rate from every starting point makes Laguerre's iteration an excellent choice for our problem.

3 Three-term recurrences

Without loss of generality, we assume the pencil (T, S) where

$$T = \begin{pmatrix} t_{11} & t_{12} & & & & \\ t_{12} & t_{22} & t_{23} & & \mathbf{0} & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & \mathbf{0} & & t_{n-2,n-1} & t_{n-1,n-1} & t_{n-1,n} \\ & & & & t_{n-1,n} & t_{nn} \end{pmatrix}, \tag{6}$$

$$S = \begin{pmatrix} s_{11} & s_{12} & & & & \\ s_{12} & s_{22} & s_{23} & & \mathbf{0} & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & \mathbf{0} & & s_{n-2,n-1} & s_{n-1,n-1} & s_{n-1,n} \\ & & & & s_{n-1,n} & s_{nn} \end{pmatrix}, \tag{7}$$

is unreducible, that is

$$t_{i,i+1}^2 + s_{i,i+1}^2 \neq 0, \quad i = 1, 2, \cdots, n-1.$$

We intend to find the eigenvalues of the pencil (T, S) by solving the the characteristic polynomial equation

$$f(\lambda) \equiv \det[T - \lambda S] = 0$$

via Laguerre's iteration (4) and (5). For this purpose, $f(\lambda)$, $f'(\lambda)$ and $f''(\lambda)$ need to be evaluated efficiently at any given $\lambda \in \boldsymbol{R}$. It is well known that the characteristic polynomial $f(\lambda)$ can be evaluated by a three term recurrence [25, p341],

$$\begin{cases} \rho_0 = 1, \quad \rho_1 = t_{11} - \lambda s_{11} \\ \rho_i = (t_{ii} - \lambda s_{ii})\rho_{i-1} - (t_{i-1,i} - \lambda s_{i-1,i})^2 \rho_{i-2}, \quad i = 2, 3, \ldots, n \end{cases} \tag{8}$$

in which $f(\lambda) = \rho_n$. As a by-product, the number of sign changes between consecutive terms of $\{\rho_i\}_{i=1}^n$, denoted by $\kappa(\lambda)$, which equals to the number of eigenvalues of (T, S)

that are less than λ [25, p341][1] is also obtained. Differentiating the recurrence (8) yields

$$\left\{\begin{array}{rcl} \rho_0' &=& 0, \quad \rho_1' = -s_{11} \\ \rho_i' &=& (t_{ii}-\lambda s_{ii})\rho_{i-1}' - s_{ii}\rho_{i-1} + 2(t_{i-1,i}-\lambda s_{i-1,i})s_{i-1,i}\rho_{i-2} \\ && \qquad - (t_{i-1,i}-\lambda s_{i-1,i})^2\rho_{i-2}' \\ && i = 2,3,\ldots,n \end{array}\right. \tag{9}$$

$$\left\{\begin{array}{rcl} \rho_0'' &=& 0, \quad \rho_1'' = 0 \\ \rho_i'' &=& (t_{ii}-\lambda s_{ii})\rho_{i-1}'' - 2s_{ii}\rho_{i-1}' - 2s_{i-1,i}^2\rho_{i-2} \\ && \qquad + 4(t_{i-1,i}-\lambda s_{i-1,i})s_{i-1,i}\rho_{i-2}' - (t_{i-1,i}-\lambda s_{i-1,i})^2\rho_{i-2}'' \\ && i = 2,3,\ldots,n \end{array}\right. \tag{10}$$

and

$$f'(\lambda) = \rho_n', \qquad f''(\lambda) = \rho_n''.$$

However, these recurrences may suffer from severe overflow and underflow problems and require frequent scaling. To remedy this, we propose the following alternative recurrences. Let

$$\xi_i = \frac{\rho_i}{\rho_{i-1}}, \quad i = 1,2,\cdots,n. \tag{11}$$

Dividing both sides of (8) by ρ_{i-1} yields,

$$\left\{\begin{array}{rcl} \xi_1 &=& t_{11} - \lambda s_{11} \\ \xi_i &=& t_{ii} - \lambda s_{ii} - \dfrac{(t_{i-1,i}-\lambda s_{i-1,i})^2}{\xi_{i-1}}, \quad i = 2,3,\cdots,n. \end{array}\right. \tag{12}$$

To avoid possible breakdown, namely, when $\xi_i = 0$ for certain i, the following adjustment is asserted:

- If $\xi_1 = 0$ (i.e. $t_{11} = \lambda s_{11}$), set $\xi_1 = t_{11}\varepsilon^2$
- If $\xi_i = 0$, $i > 1$, set $\xi_i = \dfrac{(|t_{i-1,i}|+|\lambda s_{i-1,i}|)^2\varepsilon^2}{\xi_{i-1}}$.

That is, if $\xi_i = 0$, we perturb the corresponding entries t_{11}, $t_{i-1,i}$ and/or $s_{i-1,i}$ beyond the last significant digit stored in the machine.

Obviously,

$$\rho_i = \prod_{k=1}^{i} \xi_k$$

and $\kappa(\lambda)$, the number of eigenvalues of (T,S) that are less than λ, is now equal to the number of negative terms in $\{\xi_i\}_{i=1}^n$. Let

$$\eta_i = -\frac{\rho_i'}{\rho_i} \quad \text{and} \quad \zeta_i = \frac{\rho_i''}{\rho_i} \qquad i = 0,1,\cdots,n$$

and divide (9) and (10) by ρ_i, we obtain

[1] This is true only when S is positive definite. In §6.2, we shall extend this eigenvalue counting result to the case where T is positive definite but S is not.

$$\left\{\begin{array}{rcl} \eta_0 & = & 0, \quad \eta_1 = \dfrac{s_{11}}{\xi_1} \\ \eta_i & = & \dfrac{1}{\xi_i}\,\Big[(t_{ii}-\lambda s_{ii})\eta_{i-1}+s_{ii} \\ & & \qquad -\dfrac{1}{\xi_{i-1}}\,\left(2(t_{i-1,i}-\lambda s_{i-1,i})s_{i-1,i}+(t_{i-1,i}-\lambda s_{i-1,i})^2\eta_{i-2}\right)\Big] \\ & & i=2,3,\ldots,n \end{array}\right. \tag{13}$$

and

$$\left\{\begin{array}{rcl} \zeta_0 & = & 0, \quad \zeta_1 = 0 \\ \zeta_i & = & \dfrac{1}{\xi_i}\,\Big[(t_{ii}-\lambda s_{ii})\zeta_{i-1}+2s_{ii}\eta_{i-1} \\ & & \qquad -\dfrac{1}{\xi_{i-1}}\,\left(2s_{i-1,i}^2+4(t_{i-1,i}-\lambda s_{i-1,i})s_{i-1,i}\eta_{i-2}-(t_{i-1,i}-\lambda s_{i-1,i})^2\zeta_{i-2}\right)\Big] \\ & & i=2,3,\ldots,n \end{array}\right. \tag{14}$$

where

$$-\frac{f'(\lambda)}{f(\lambda)}=\eta_n, \qquad \frac{f''(\lambda)}{f(\lambda)}=\zeta_n$$

which are what we need in Laguerre's iteration. Recurrences (12), (13) and (14) virtually eliminate the hazards of overflow and underflow problems because of their self-scaling.

Remark 1: In recurrences (12), (13) and (14), the positive definiteness of S is irrelevant. Switching the roles of T and S in these recurrences causes no extra difficulties. So T and S can be treated in a symmetric manner. That is, to evaluate the zeros of $f(\lambda)\equiv\det[T-\lambda S]$, we may evaluate the zeros of $g(\mu)\equiv\det[\mu T-S]$ instead, acknowledging the relationship $\lambda=1/\mu$. Notice that when S is positive definite, μ can not be zero.

Remark 2: The transformation similar to (11) has been used for bisection code BISECT in EISPACK [21] for the standard symmetric tridiagonal eigenvalue problem.

4 Error analysis

Obviously,

$$f(\lambda)\equiv\det[T-\lambda S]=\prod_{i=1}^{n}\xi_i$$

where ξ_i, $i=1,2,\cdots,n$ are evaluated by the recurrence (12) at λ. Let $fl(\bullet)$ denote the floating point computation of $\bullet$. In practice, when we intend to evaluate the zeros of $f(\lambda)$, actually the zeros of $fl[f(\lambda)]$ are evaluated. We shall prove that, zeros of $fl[f(\lambda)]$ are exact eigenvalues of a nearby symmetric definite pencil $(T+\delta T, S+\delta S)$ with small δT and δS, and thereby establish the backward stability of our algorithm.

The most frequently used model for the floating point arithemetic is

$$fl(x\circ y)=(x\circ y)\cdot(1+e), \quad |e|\le\varepsilon,$$

where $\circ$ is either $+$, $-$ $\times$ or $/$. For computers without a guard digit, such as Cray, floating point addition and subtraction satisfy

$$fl(x \pm y) = x(1+e_1) \pm y(1+e_2), \quad |e_i| \leq \varepsilon, \quad i = 1,2$$

instead. The ε above is the machine precision, which is approximately 6×10^{-8} for single precision and 2.2×10^{-16} for double precision under IEEE standard. If underflow is taken into consideration, the model becomes

$$fl(x \circ y) = (x \circ y) \cdot (1+e) + e_u, \quad |e| \leq \varepsilon, \quad |e_u| \leq \varepsilon_u, \quad e \cdot e_u = 0, \tag{15}$$

for machines with a guard digit. For machines without a guard digit, adjustment on addition and subtraction can be made as follows:

$$fl(x \pm y) = x(1+e_1) \pm y(1+e_2) + e_u, \quad |e_i| \leq \varepsilon, \quad i = 1,2, \quad |e_u| \leq \varepsilon_u, \quad e_u(e_1^2 + e_2^2) = 0 \tag{16}$$

where e_u is the underflow threshold, which is approximately 10^{-38} for single precision and 10^{-308} for double precision under IEEE standard. In the following error analysis, we assume the most general models (15) for $\times$ and $/$, and (16) for $\pm$. Other models are special cases. Under these models, we also have

$$\begin{aligned} fl(x^2) &= (x+\tilde{e}_u)^2(1+e) = [(x+\tilde{e}_u)(1+e')]^2 = (x+e'')^2 \qquad (17) \\ & \quad |e''| \leq \frac{1}{2}|x|\varepsilon + \sqrt{\varepsilon_u} + O(\varepsilon^2) \\ & \quad |e| \leq \varepsilon, \quad |e'| \leq \frac{1}{2}\varepsilon + O(\varepsilon^2), \quad |\tilde{e}_u| \leq \sqrt{\varepsilon_u}, \quad e \cdot \tilde{e}_u = 0. \end{aligned}$$

Proposition 4.1

$$fl[f(\lambda)] = (1+\gamma)\det[(T+\delta T) - \lambda(S+\delta S)] \tag{18}$$

where

$$|\gamma| \leq n\varepsilon,$$

both δT and δS are symmetric tridiagonal matrices satisfying entrywise inequalities

$$|\delta T| \leq 2.51\varepsilon|T| + \sqrt{\varepsilon_u}, \quad |\delta S| \leq 3.51\varepsilon|S|. \tag{19}$$

Proof: By the model of floating point arithmetic

$$\begin{aligned} fl(\xi_1) &= fl(t_{11} - \lambda s_{11}) \\ &= t_{11}(1+e_{11}) - fl(\lambda s_{11})(1+e_{12}) + e_{u_{11}} \\ &= t_{11}(1+e_{11}) - (\lambda s_{11}(1+e_{13}) + e_{u_{12}})(1+e_{12}) + e_{u_{11}} \\ &= (t_{11} + \tau_{11}) - \lambda(s_{11} + \sigma_{11}) \end{aligned}$$

where

$$|\tau_{11}| = |e_{11}t_{11} + e_{u_{11}} + e_{u_{12}}(1+e_{12})| \leq \varepsilon|t_{11}| + 2\varepsilon_u + O(\varepsilon^2) \tag{20}$$

$$|\sigma_{11}| = |e_{13}s_{11}| \leq \varepsilon|s_{11}|. \tag{21}$$

Also,

$$\begin{aligned}
fl(t_{ii}-\lambda s_{ii}) &= t_{ii}(1+e_{i1})-(\lambda s_{ii}(1+e_{i3})+e_{u_{i2}})(1+e_{i2})+e_{u_{i1}} \\
&= [t_{ii}+(e_{i1}t_{ii}+e_{u_{i1}}+e_{u_{i2}}(1+e_{i2}))] \\
&\qquad -\lambda[s_{ii}+(e_{i2}+e_{i3}+e_{i2}e_{i3})s_{ii}] \\
fl((t_{i-1,i}-\lambda s_{i-1,i})^2) &= \{fl[t_{i-1,i}-\lambda s_{i-1,i})+\tilde{e}_{u_{i3}}](1+e''_{i4})\}^2 \\
&= \{[(t_{i-1,i}+(e_{i5}t_{i-1,i}+e_{u_{i4}}+e_{u_{i5}}(1+e_{i6}))+\tilde{e}_{u_{i3}}) \\
&\qquad -\lambda(s_{i-1,i}+(e_{i7}+e_{i8}+e_{i7}e_{i8})s_{i-1,i})](1+e''_{i4})\}^2
\end{aligned}$$

Thus

$$\begin{aligned}
fl(\xi_i) &= fl\left((t_{ii}-\lambda s_{ii})-\frac{(t_{i-1,i}-\lambda s_{i-1,i})^2}{\xi_{i-1}}\right) \\
&= fl(t_{ii}-\lambda s_{ii})(1+e_{i8}) \\
&\qquad -\left(\frac{fl((t_{i-1,i}-\lambda s_{i-1,i})^2)}{fl(\xi_{i-1})}(1+e_{i9})+e_{u_{i4}}\right)(1+e_{i10}) \\
&= (t_{ii}+\tau_{ii})-\lambda(s_{ii}+\sigma_{ii})-\frac{[(t_{i-1,i}+\tau_{i-1,i})-\lambda(s_{i-1,i}+\sigma_{i-1,i})]^2}{fl(\xi_{i-1})}
\end{aligned}$$

where

$$\begin{aligned}
|\tau_{ii}| &\le 2.0\varepsilon|t_{ii}|+3\varepsilon_u+O(\varepsilon^2) \\
|\tau_{i-1,i}| &\le 2.5\varepsilon|t_{i-1,i}|+\sqrt{\varepsilon_u}+O(\varepsilon^2) \\
|\sigma_{ii}| &\le 3.0\varepsilon|s_{ii}|+O(\varepsilon^2) \\
|\sigma_{i-1,i}| &\le 3.5\varepsilon|s_{i-1,i}|+O(\varepsilon^2)
\end{aligned}$$

That is, $fl(\xi_i)$, $i=1,2,\cdots,n$ is the *exact* recurrence in (12) for the matrix pencil $(T+\delta T, S+\delta S)$ where $\delta T=(\tau_{ij})$ and $\delta S=(\sigma_{ij})$ satisfying (19). Now,

$$\begin{aligned}
fl[f(\lambda)] &= fl\left(\prod_{i=1}^{n} fl(\xi_i)\right) \\
&= \prod_{j=1}^{n-1}(1+\varepsilon_j)\prod_{i=1}^{n} fl(\xi_i) \\
&= (1+\gamma)\prod_{i=1}^{n} fl(\xi_i) \\
&= (1+\gamma)\det[(T+\delta T)-\lambda(S+\delta S)]
\end{aligned}$$

where $|\gamma|\le(n-1)\varepsilon+O(\varepsilon^2)$. ■

In actual computation, the eigenvalue counter $\kappa(\lambda)$ is obtained by counting the number of negative terms of $\{fl(\xi_i)\}_{i=1}^n$. Therefore the argument in the proof above also provides the accuracy of $\kappa(\lambda)$.

Corollary 4.2 *Let $fl(\kappa(\lambda))$ be the number of negative terms in the sequence $\{fl(\xi_i)\}_{i=1}^n$ obtained from (12) using floating point operations. Then $fl(\kappa(\lambda))$ is the number of eigenvalues of the pencil $(T+\delta T, S+\delta S)$ in the interval $(-\infty,\lambda)$, where δT and δS satisfy (19).*

Corollary 4.3 *Let δT and δS be as in Proposition 4.1. Let S be a positive definite matrix with eigenvalues $\lambda_1(S) \leq \lambda_2(S) \leq \cdots \leq \lambda_n(S)$ such that $\lambda_1(S) > 3.51\varepsilon\|S\|_\infty$. Then $(T+\delta T, S+\delta S)$ is a positive definite pencil.*

Proof: Let $\lambda_{min}(S+\delta S)$ be the smallest eigenvalue of $S+\delta S$. By standard perturbation theory [8, page 411],

$$\lambda_{min}(S+\delta S) > \lambda_1(S) - \|\delta S\|_\infty \geq \lambda_1(S) - 3.51\varepsilon\|S\|_\infty > 0.$$

■

This corollary implies that, unless the pencil (T, S) is extremely ill-conditioned for which $\lambda_1 < 3.51\varepsilon\|S\|_\infty$, the floating point computation of our algorithm is still performed on a symmetric definite pencil.

The above analysis establishes the backward stability of our algorithm. The actual errors in computed eigenvalues depend on the condition of the pencil (T, S).

Definition 4.4 *The number*

$$c(A,B) = \min_{\|x\|_2=1} \left[(x^\mathsf{T}Ax)^2 + (x^\mathsf{T}Bx)^2\right]$$

is called the Crawford number *of the pencil (A, B).*

Theorem 4.5 *[8, page 468] Suppose (A, B) is an $n \times n$ symmetric definite pencil with eigenvalues*

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

Suppose δA and δB are symmetric $n \times n$ matrices satisfying

$$\epsilon^2 = \|\delta A\|_2^2 + \|\delta B\|_2^2 < c(A,B).$$

Then the pencil $(A+\delta A, B+\delta B)$ is a symmetric definite pencil with eigenvalues $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_n$ which satisfy

$$|\arctan(\lambda_i) - \arctan(\mu_i)| \leq \arctan\left(\frac{\epsilon}{c(A,B)}\right)$$

for $i = 1, 2, \cdots, n$.

From Proposition 4.1 and Theorem 4.5, for our symmetric definite pencil (T, S) with spectrum $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and inequality

$$\sigma^2 \equiv (2.51\varepsilon\|T\|_\infty + \sqrt{\varepsilon_u})^2 + (3.51\varepsilon\|S\|_\infty)^2 < c(T,S),$$

we have

$$|\arctan(\lambda_i) - \arctan(\mu_i)| \leq \arctan\left(\frac{\sigma}{c(T,S)}\right)$$

for $i = 1, 2, \cdots, n$, where $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_n$ are zeros of $fl(f(\lambda))$.

5 Splitting the pencil

Let T and S be as in (6) and (7) respectively. By choosing $k \approx n/2$ and setting $s_{k,k+1} = t_{k,k+1} = 0$, we obtain a split pencil $(\hat{T}, \hat{S})$, where

$$\hat{T} = \begin{pmatrix} T_0 & 0 \\ 0 & T_1 \end{pmatrix}, \quad \hat{S} = \begin{pmatrix} S_0 & 0 \\ 0 & S_1 \end{pmatrix} \tag{22}$$

and

$$T_0 = \begin{bmatrix} t_{11} & t_{12} & & \\ t_{12} & \ddots & \ddots & \\ & \ddots & \ddots & t_{k-1,k} \\ & & t_{k-1,k} & t_{k,k} \end{bmatrix}, \quad T_1 = \begin{bmatrix} t_{k+1,k+1} & t_{k+1,k+2} & & \\ t_{k+1,k+2} & \ddots & \ddots & \\ & \ddots & \ddots & t_{n-1,n} \\ & & t_{n-1,n} & t_{nn} \end{bmatrix}$$

$$S_0 = \begin{bmatrix} s_{11} & s_{12} & & \\ s_{12} & \ddots & \ddots & \\ & \ddots & \ddots & s_{k-1,k} \\ & & s_{k-1,k} & s_{k,k} \end{bmatrix}, \quad S_1 = \begin{bmatrix} s_{k+1,k+1} & s_{k+1,k+2} & & \\ s_{k+1,k+2} & \ddots & \ddots & \\ & \ddots & \ddots & s_{n-1,n} \\ & & s_{n-1,n} & s_{nn} \end{bmatrix}.$$

Proposition 5.1 *Both (T_0, S_0) and (T_1, S_1) are symmetric definite pencils.*

Proof: T_i, S_i, $i = 0, 1$ are obviously symmetric. S_0 and S_1 are positive definite because S is positive definite and so are principle submatrices of any permutations of S. ∎

Theorem 5.2 *Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be eigenvalues of the pencil (T, S) and $\hat{\lambda}_1 \leq \hat{\lambda}_2 \leq \cdots \leq \hat{\lambda}_n$ be eigenvalues of the pencil $(\hat{T}, \hat{S})$. Then,*

$$\begin{array}{rcl} -\infty < & \lambda_1 & \leq \hat{\lambda}_1 \\ \hat{\lambda}_{i-1} \leq & \lambda_i & \leq \hat{\lambda}_{i+1}, \quad i = 2, 3, \cdots, n-1 \\ \hat{\lambda}_n \leq & \lambda_n & < \infty. \end{array}$$

To prove the above theorem, let's consider the parameterized pencil

$$(T(\alpha), S(\alpha)) = ((1-\alpha)\hat{T} + \alpha T, (1-\alpha)\hat{S} + \alpha S), \quad \alpha \in [0, 1].$$

Apparently, $T(\alpha)$ equals to T entrywise except $t_{k,k+1}$ of T is replaced by $\alpha t_{k,k+1}$ in $T(\alpha)$. Similarly, $S(\alpha)$ can be obtained by changing $s_{k,k+1}$ of S to $\alpha s_{k,k+1}$. Also, $(T(0), S(0)) = (\hat{T}, \hat{S})$ and $(T(1), S(1)) = (T, S)$.

Lemma 5.3 *$(T(\alpha), S(\alpha))$ is a symmetric definite pencil for each $\alpha \in [0, 1]$.*

Proof: Both $T(\alpha)$ and $S(\alpha)$ are clearly symmetric. For each $\alpha \in [0, 1]$, let $\mu(\alpha)$ be the smallest eigenvalue of $S(\alpha)$. Then $\mu(\alpha)$ is a monotone function [12] with $\mu(1) \leq \mu(0)$ by the interlacing property of symmetric matrices [8, page 411]. Thus $\mu(\alpha) \geq \mu(1) > 0$, So, $S(\alpha)$ is positive definite for each $\alpha \in [0, 1]$. ∎

Corollary 5.4 *For each $\alpha \in [0,1]$, the pencil $(T(\alpha), S(\alpha))$ has n real eigenvalues.*

Let

$$\lambda_1(\alpha) \le \lambda_2(\alpha) \le \cdots \le \lambda_n(\alpha)$$

be eigenvalues of the pencil $(T(\alpha), S(\alpha))$, $\alpha \in [0,1]$. Then each $\lambda_i(\alpha)$ is a continuous function of α in $[0,1]$. This follows from the continuity of eigenvalues with respect to the entries of the pencil.

Proof of Theorem 5.2. When a standard three term recurrence, such as (8), for calculating the determinant of a symmetric tridiagonal matrix is applied to

$$T(\alpha) - \lambda S(\alpha) = \left(\begin{array}{cccc} \ddots & \ddots & & \\ \ddots & t_{kk} - \lambda s_{kk} & \alpha(t_{k,k+1} - s_{k,k+1}) & \\ & \alpha(t_{k,k+1} - s_{k,k+1}) & t_{k+1,k+1} - s_{k+1,k+1} & \ddots \\ & & \ddots & \ddots \end{array} \right)$$

one can easily see that the variable α appears in $\det[T(\alpha) - \lambda S(\alpha)]$ only in the second degree. Namely,

$$H(\alpha, \lambda) \equiv \det[T(\alpha) - \lambda S(\alpha)] = p(\lambda) + \alpha^2 q(\lambda) \qquad (23)$$

for certain polynomials $p(\lambda)$ and $q(\lambda)$. In fact, $p(\lambda) = \det[\hat{T} - \lambda \hat{S}]$. If $q(\lambda_0) \neq 0$ for certain λ_0, then $H(\alpha, \lambda_0)$, considered as a polynomial in α, can have at most one zero in $[0,1]$.

Now, by the interlacing property [25, p340], we have

$$\lambda_1 \le \hat{\lambda}_1 \quad \text{and} \quad \hat{\lambda}_n \le \lambda_n.$$

Suppose $\lambda_i < \hat{\lambda}_{i-1}$ for some $i \in \{2, 3, \cdots, n-1\}$, or $\lambda_i(1) < \lambda_{i-1}(0)$. Since $\lambda_{i-1}(1) \le \lambda_i(1) < \lambda_{i-1}(0) \le \lambda_i(0)$, by the continuity of both $\lambda_{i-1}(\alpha)$ and $\lambda_i(\alpha)$, for any $\lambda_0 \in (\lambda_i(1), \lambda_{i-1}(0))$ there exist α_{i-1} and α_i in $(0,1)$ such that $\lambda_i(\alpha_i) = \lambda_{i-1}(\alpha_{i-1}) = \lambda_0$. Or, $H(\alpha, \lambda_0) = 0$ in (23) has at least two solutions for α in $(0,1)$. This can only happen when $q(\lambda_0) = 0$. But then, $H(\alpha, \lambda_0) = p(\lambda_0) = \det[\hat{T} - \lambda_0 \hat{S}] = 0$ implies that λ_0 is an eigenvalue of the pencil $(\hat{T}, \hat{S})$. A contradiction is achieved since $(\hat{T}, \hat{S})$ can have at most finitely many eigenvalues. So $\lambda_i \ge \hat{\lambda}_{i-1}$. By a similar argument, one can prove $\lambda_i \le \hat{\lambda}_{i+1}$, $i = 2, 3, \cdots, n-1$. ■

By Theorem 5.2, each eigenvalue λ_i of the pencil (T, S) is next to its corresponding eigenvalue $\hat{\lambda}_i$ of the split pencil $(\hat{T}, \hat{S})$. Thus $\hat{\lambda}_i$ can always be used as a starting point of Laguerre's iteration for finding λ_i.

6 The algorithm

6.1 Merging the pencil in an interval $[a, b)$

In this section we describe our basic algorithm which evaluates eigenvalues of the pencil (T, S) in a specified interval $[a, b)$. If this interval is large enough, all eigenvalues will be included. In many applications, only a fraction of all the eigenvalues are needed. Our algorithm is suitable for this partial spectrum evaluation. We shall discuss the full spectrum evaluation in §6.2.

Finding eigenvalues of the pencil (T, S) in $[a, b)$ consists of three steps:

1. *Split* the pencil (T, S) into a reduced pencil $(\hat{T}, \hat{S}) = \left(\begin{pmatrix} T_0 & 0 \\ 0 & T_1 \end{pmatrix}, \begin{pmatrix} S_0 & 0 \\ 0 & S_1 \end{pmatrix} \right)$, where the subpencils (T_0, S_0) and (T_1, S_1) are about the same size.

2. Evaluate eigenvalues of the reduced pencil $(\hat{T}, \hat{S})$ in the interval $[a, b)$. Namely, evaluate eigenvalues of subpencils (T_0, S_0) and (T_1, S_1) in the interval $[a, b)$.

3. *Merge* the subpencils (T_0, S_0) and (T_1, S_1) back to (T, S). More precisely, evaluate eigenvalues of (T, S) in the interval $[a, b)$ by using Laguerre's iteration to solve the equation $f(\lambda) \equiv \det[T - \lambda S] = 0$ starting from the results of step 2.

These three steps are executed recursively. When the pencil (T, S) is split into two subpencils (T_0, S_0) and (T_1, S_1), both subpencils are still symmetric tridiagonal definite pencils and can be further split into smaller subpencils until the size become 2×2 or 1×1. So without loss of generality, we may assume that eigenvalues of subpencils (T_0, S_0) and (T_1, S_1) are known and only consider step 3, the merging step, in the following.

6.1.1 Starting points for Laguerre's iteration

Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be eigenvalues of (T, S). The recurrence (12) provides an eigenvalue counter $\kappa(\lambda)$ which represents the number of eigenvalues of (T, S) that are less than λ. So, let $k_1 = \kappa(a)$ and $p = \kappa(b) - \kappa(a)$ then

$$\lambda_{k_1+1} \leq \lambda_{k_1+2} \leq \cdots \leq \lambda_{k_1+p}$$

are all the eigenvalues of (T, S) in $[a, b)$. Let

$$\hat{\lambda}_{k_2+1} \leq \hat{\lambda}_{k_2+2} \leq \cdots \leq \hat{\lambda}_{k_2+m}$$

be all eigenvalues of the split pencil $(\hat{T}, \hat{S})$ in $[a, b)$. By Theorem 5.2,

$$\lambda_{k_2-1} \leq \hat{\lambda}_{k_2} < a, \qquad b \leq \hat{\lambda}_{k_2+m+1} \leq \lambda_{k_2+m+2}$$

and

$$a \leq \hat{\lambda}_{k_2+1} \leq \lambda_{k_2+2}, \qquad \lambda_{k_2+m-1} \leq \hat{\lambda}_{k_2+m} < b.$$

It follows that there are at least $m - 2$ and at most $m + 2$ eigenvalues of (T, S) in $[a, b)$. (The maximum possible set of those eigenvalues is $\lambda_{k_2} \leq \lambda_{k_2+1} \leq \cdots \leq \lambda_{k_2+m+1}$.) Therefore, $k_1 \leq k_2 \leq k_1 + 1$ and $|p - m| \leq 2$. Let

$$\mu_{k_2} = a, \;\; \mu_{k_2+1} = \hat{\lambda}_{k_2+1}, \cdots, \;\; \mu_{k_2+m} = \hat{\lambda}_{k_2+m}, \;\; \mu_{k_2+m+1} = b.$$

Then $\{\mu_{k_1+1}, \mu_{k_1+2}, \cdots, \mu_{k_1+p}\}$ is a proper set of starting points of Laguerre's iteration to evaluate corresponding eigenvalues $\lambda_{k_1+1}, \cdots, \lambda_{k_1+p}$ of (T, S) in $[a, b)$.

6.1.2 Bisection or Laguerre's iteration

Let

$$\begin{aligned}\mathcal{I}_k &= [\mu_{k_2}, \mu_{k_2+1}],\\ \mathcal{I}_l &= [\mu_{l-1}, \mu_{l+1}], \quad l = k_2+1, \cdots, k_2+m,\\ \mathcal{I}_{k+m+1} &= [\mu_{k_2+m}, \mu_{k_2+m+1}].\end{aligned}$$

Apparently, for target eigenvalues λ_j, $j = k_1+1, \cdots k_1+p$ of (T, S) in $[a, b)$, we have $\lambda_j \in \mathcal{I}_j$ for each j. However, the (computed) interval $\mathcal{I}_j$ may not contain λ_j because of the error in floating point operation. When this happens, the end point of $\mathcal{I}_j$ closer to λ_j will be accepted as approximate value of λ_j. By this strategy, the *approximate* value of λ_j will always be considered being in $\mathcal{I}_j$.

To find λ_j, $j = k_1+1, \cdots, k_1+p$, we start with an interval $\mathcal{I}_j$ which contains λ_j and a starting point μ_j. Write $\mathcal{I}_j \equiv [a_j, b_j]$. Laguerre's iteration is cubically convergent under the condition that the $+/-$ sign in $L_\pm$ or $L_{r\pm}$ (see (4) and (5)) agree with the sign of $-\dfrac{f'}{f}$ [25, p445]. This agreement in sign can always be achieved in a neighborhood of each zero of f. In general, to approximate λ_j from x, the sign in $L_\pm$ or $L_{r\pm}$ must be chosen as $sign\,(\lambda_j - x)$ because of the monotonicity of Laguerre's iteration. (The sign of $\lambda_j - x$ can be determined by $\kappa(x)$.) So, the requirement $sign\left(-\dfrac{f'(x)}{f(x)}\right) = sign\,(\lambda_j - x)$ dictates our decision to perform Laguerre's iteration. When this requirement fails, the bisection is used instead until the sign agreement is achieved.

More specifically, at x in the starting interval $\mathcal{I}_j = [a_j, b_j]$, the following process is taken before starting Laguerre's iteration.

```
(#) use (12), (13) and (14) at $x$ to obtain $-\dfrac{f'(x)}{f(x)}$ , $\dfrac{f''(x)}{f(x)}$ and $\kappa(x)$
    update $[a_j, b_j]$ according to $\kappa(x)$, i.e.
        $(a_j, b_j) = \begin{cases} (x, b_j) & \text{if } x < \lambda_j \text{ or } \kappa(x) < j \\ (a_j, x) & \text{if } x > \lambda_j \text{ or } \kappa(x) \geq j \end{cases}$
    if $sign\left(-\dfrac{f'(x)}{f(x)}\right) \neq sign(\lambda_j - x)$ then
        $x = (a_j + b_j)/2$
        go to (#)
    end if
```

6.1.3 Estimating multiplicity of the target eigenvalue

We use the algorithm EstMlt in [14] to estimate the multiplicity of the target eigenvalue λ_j. As indicated in [14], an overestimate of the multiplicity causes no trouble and can be dynamically adjusted. The algorithm EstMlt is listed in Figure 1.

6.1.4 Stopping criteria for Laguerre's iteration

Once Laguerre's iteration is used, we proceed the process until virtually no further improvement can be achieved. Suppose $x_1, x_2, \cdots, x_k$ have been obtained from Laguerre's

```
Algorithm EstMlt

    input:  initial point $x$, $sign(\lambda_j - x)$, subscript $j$,
                eigenvalues $\hat\lambda_{k_1} < \cdots < \hat\lambda_{k_2}$ of pencil $(\hat T, \hat S)$
    output: $mlt$, the estimated numerical multiplicity of
                the eigenvalue $\lambda_j$ of $(T, S)$
    begin EstMlt
        $mlt = 1$
        for $k = 1, 2, \cdots$
            $m = j + k \cdot sign(\lambda_j - x)$
            if $|\hat\lambda_j - \hat\lambda_m| < 0.01|x - \hat\lambda_j|$ then
                $mlt = mlt + 1$
            else
                go to (#)
            end if
        end for
(#) end EstMlt
```

Figure 1: **Algorithm** EstMlt

iteration. Let ε be the machine precision. We accept x_k as an approximate zero x^* of $f(\lambda)$ if

$$\left\{\begin{array}{lll} \text{(i)} & |x_k - x_{k-1}| \le \varepsilon |x_k|; & \text{or} \\ \text{(ii)} & |x_k - x_{k-1}| \ge |x_{k-1} - x_{k-2}|; & \text{or} \\ \text{(iii)} & \frac{|x_k - x_{k-1}|^2}{|x_{k-1} - x_{k-2}| - |x_k - x_{k-1}|} \le \varepsilon |x_k|. & \end{array}\right. \tag{24}$$

In (24) above, (i) implies that the improvement on x_{k-1} is beyond its last digit stored in the computer, which is desirable but it may be difficult to achieve when x^* has a tiny magnitude compared to $\|T\|_\infty$ and $\|S\|_\infty$. (iii) is proposed by Kahan [10]. The left-hand side of (iii) is an estimate of $|x_k - x^*|$ assuming at least linear convergence occurs. When (iii) is observed, the accuracy improvement of x_{k+1} from x_k is expected to be beyond the last digit of x_k. If both (i) and (iii) fail to happen, eventually (ii) will occur which indicates that no further improvement can be obtained by Laguerre's iteration using floating point arithematic.

6.1.5 The actual Laguerre's iteration process LagIt

We adopt the algorithm LagIt in [14] with some modifications. The algorithm, shown in Figure 2, is based on Laguerre's iteration formula (5) with bisection as backup and dynamic adjustments in estimating the multiplicity.

The algorithm Merge which evaluates eigenvalues of the pencil (T, S) in $[a, b)$, starting from eigenvalues of the split pencil $(\hat T, \hat S)$ in the same interval $[a, b)$ is given in Figure 3.

```
Algorithm LagIt
    Input:  subscript $j$, initial point $x \in (\lambda_{j-1}, \lambda_{j+1})$,
                interval $\mathcal{I}_j = [a_j, b_j] \ni \lambda_j$, $mlt$ evaluated by EstMlt
    Output: the $j$-th eigenvalue $\lambda_j$ of the pencil $(T, S)$.
    begin LagIt
        $x_1 = x$
        for $l = 2, 3, \cdots$
            $x_l = \begin{cases} L_{r+}(x_{l-1}), & \text{if } \kappa(x_{l-1}) < j \\ L_{r-}(x_{l-1}), & \text{if } \kappa(x_{l-1}) \geq j \end{cases}$  with $r = mlt$
            $\delta_l = x_l - x_{l-1}$
            if (24) is satisfied, go to (#)
    (##)    evaluate $-\frac{f'(x_l)}{f(x_l)}$, $\frac{f''(x_l)}{f(x_l)}$ and $\kappa(x_l)$ using (12), (13) and (14)
            update $[a_j, b_j]$ according to $\kappa(x_l)$
            if $mlt > 1$ and $|\kappa(x_l) - \kappa(x_{l-1})| > 1$ then
                $mlt = |\kappa(x_l) - \kappa(x_{l-1})|$,   $x_l = (x_l + x_{l-1})/2$
                go to (##)
            end if
        end for
(#)     $\lambda_i = x_l$
    end LagIt
```

Figure 2: **Algorithm** LagIt

6.2 Evaluating all the eigenvalues of (T, S)

All eigenvalues of (T, S) are in the interval $[-\|S^{-1}T\|, \|S^{-1}T\|]$ [17, p308]. However, when S is nearly singular, $\|S^{-1}T\|$ can be arbitrarily large. It is improper to evaluate $\|S^{-1}T\|$ and carry out the computation in a large interval. An alternative is to switch the roles of T and S and restrict all the computations in $[-1, 1)$.

If $\lambda \neq 0$ is an eigenvalue of (T, S), then $1/\lambda$ is an eigenvalue of the pencil (S, T) and $-1/\lambda$ is an eigenvalue of $(S, -T)$. So, to evaluate all eigenvalues of (T, S), we may,

(i) *Evaluate eigenvalues of* (T, S) *in* $[-1, 1 + \varepsilon)$.

(ii) *Evaluate eigenvalues of* (S, T) *in* $[0, 1)$. The reciprocals of those values are eigenvalues of (T, S) in $(1, +\infty)$. Notice that 0 can not be an eigenvalue of (S, T).

(iii) *Evaluate eigenvalues of* $(S, -T)$ *in* $[0, 1)$. The negative reciprocals of these values are eigenvalues of (T, S) in $(-\infty, -1)$. Again, 0 can not be an eigenvalue of $(S, -T)$.

The eigenvalue counter $\kappa(\lambda)$, defined in §3, plays an important role in our algorithm. It is the only part in the algorithm which requires S, the second matrix in the pencil, to be positive definite, and in such cases, $\kappa(\lambda)$ equals to the number of eigenvalues of (T, S) which are less than λ. In (ii) and (iii) above, neither T nor $-T$ is assumed to be positive definite. To show the number $\kappa(\lambda)$ still serves as an effective eigenvalue counter

```
Algorithm MERGE

    input:  pencil $(T,S)$ as in (6) and (7), interval $[a,b)$
            eigenvalues $\hat{\lambda}_{k+1} < \cdots < \hat{\lambda}_{k+m}$ of the split pencil $(\hat{T},\hat{S})$
            (see (22) ) in $[a,b)$.
    Output:  eigenvalues of $(T,S)$ in $[a,b)$.
    begin MERGE
        set $\hat{\lambda}_k = a$, $\hat{\lambda}_{k+m+1} = b$, $k_1 = \kappa(a)+1$, $k_2 = \kappa(b)$
        for $j = k_1 : k_2$
            determine the interval $\mathcal{I}_j = (a_j, b_j) \ni \lambda_j$ according to §6.1.2
            if $|b_j - a_j| > \max\{|a_j|, |b_j|\}\,\varepsilon$ then
                $x = \hat{\lambda}_j$
    (#)         evaluate $-\frac{f'(x)}{f(x)}$ , $\frac{f''(x)}{f(x)}$ and $\kappa(x)$ by (12), (13) and (14)
                $(a_j, b_j) = \begin{cases} (x, b_j) & \text{if } x < \lambda_j \text{ (i.e. } \kappa(x) < j) \\ (a_j, x) & \text{if } x > \lambda_j \text{ (i.e. } \kappa(x) \ge j) \end{cases}$
                if $\kappa(x) \notin \{j-1, j\}$ or $sign\left(-\frac{f'(x)}{f(x)}\right) \neq sign(\lambda_j - x)$ then
                    $x = (a_j + b_j)/2$
                    go to (#)
                end if
                call ESTMLT to estimate the multiplicity of $\lambda_j$
                call LAGIT to obtain $\lambda_j$
            else
                $\lambda_j = (a_j + b_j)/2$
            end if
        end for
    end MERGE
```

Figure 3: **Algorithm** MERGE

in those cases, let $\kappa_{(A,B)}(\lambda)$ denote the number of negative terms in the sequence $\{\xi_i\}_{i=1}^n$ obtained from (12) with (T,S) replaced by (A,B).

In (ii), for λ in $(0,1)$, λS is positive definite. On the other hand,

$$S - \lambda T = (-\lambda T) - \left(-\frac{1}{\lambda}\right)(\lambda S),$$

so,

$$\begin{aligned} \kappa_{(S,T)}(\lambda) &= \kappa_{(-\lambda T, \lambda S)}\left(-\frac{1}{\lambda}\right) \\ &= \text{the number of eigenvalues of } (-\lambda T, \lambda S) \text{ which are less than } -\frac{1}{\lambda} \\ &\qquad \text{(because } \lambda S \text{ is positive definite)} \\ &= \text{the number of eigenvalues of } (-T, S) \text{ which are less than } -\frac{1}{\lambda} \end{aligned}$$

$$\begin{aligned}
& \text{(because } (-\lambda T, \lambda S) \text{ and } (-T, S) \text{ have the same eigenvalues)} \\
= \quad & \text{the number of eigenvalues of } (T, S) \text{ which are larger than } \frac{1}{\lambda} \\
= \quad & \text{the number of eigenvalues of } (S, T) \text{ which are less than } \lambda.
\end{aligned}$$

Similarly, in (iii), λS is positive definite for $\lambda \in (0, 1)$ and

$$S - \lambda(-T) = \lambda T - \left(-\frac{1}{\lambda}\right)(\lambda S)$$

implies,

$$\begin{aligned}
\kappa_{(S,-T)}(\lambda) \quad = \quad & \kappa_{(\lambda T, \lambda S)}\left(-\frac{1}{\lambda}\right) \\
= \quad & \text{the number of eigenvalues of } (\lambda T, \lambda S) \text{ which are less than } -\frac{1}{\lambda} \\
& \text{(because } \lambda S \text{ is positive definite)} \\
= \quad & \text{the number of eigenvalues of } (T, S) \text{ which are less than } -\frac{1}{\lambda} \\
& \text{(because } (\lambda T, \lambda S) \text{ and } (T, S) \text{ have the same eigenvalues)} \\
= \quad & \text{the number of eigenvalues of } (S, T) \text{ which are larger than } -\lambda \\
= \quad & \text{the number of eigenvalues of } (S, -T) \text{ which are less than } \lambda.
\end{aligned}$$

6.3 Split–merge algorithm

With the mechanisms developed in this section, we can summarize our algorithm as follows. Given a generalized eigenproblem (1) of the symmetric tridiagonal definite pencil (T, S) in (6) and (7). We split the pencil (T, S) into subpencils (T_0, S_0) and (T_1, S_1) of approximately equal size. Each subpencil is a symmetric tridiagonal definite pencil and is thus further split in the same manner into two smaller sub-subpencils of similar sizes. This process is continued until 1×1 or 2×2 pencils are obtained whose eigenvalues are trivial. A binary tree is constructed in Fig 4. We then reverse the direction from those 1×1 and 2×2 pencils and merge every two subpencils into a larger pencil by using Laguerre's iteration starting from eigenvalues of subpencils to find eigenvalues of the larger pencil. This merging process is continued until we eventually merge (T_0, S_0) and (T_1, S_1) into (T, S) (see Fig 4). If only eigenvalues in an interval are required, the merging process is restricted in that interval, namely, at every stage of merging, only eigenvalues in that interval are evaluated using the algorithm Merge so that we can take full advantage of partial specturm evaluation.

7 Numerical results

In this section we present preliminary experiments designed to test the performance of our algorithm ZRST and the comparison with other algorithms.

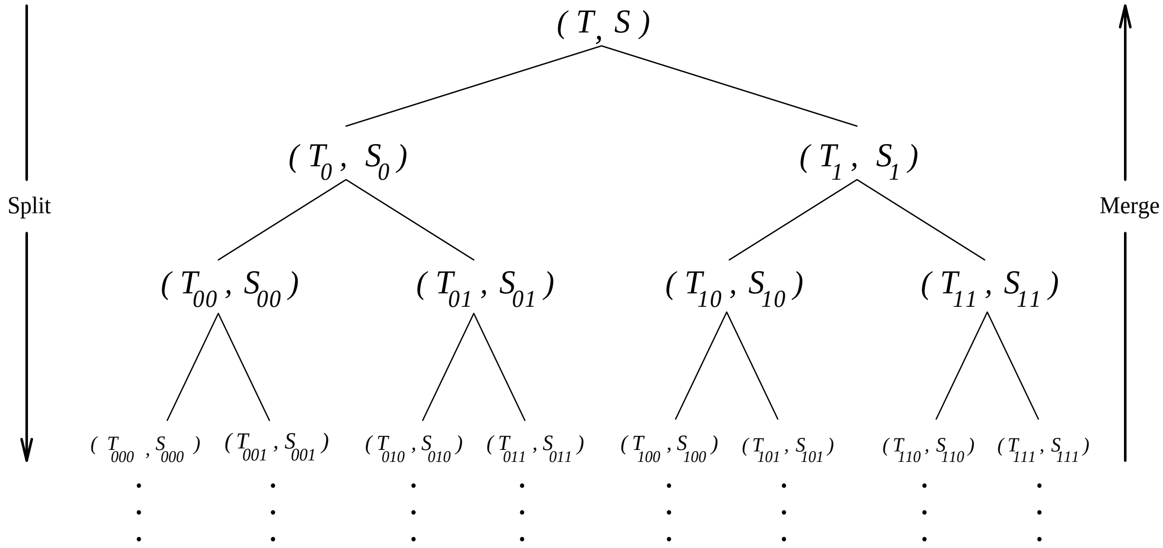

Figure 4: Split-merge algorithm

Experiment 1. We implemented our algorithm ZSRT and the EISPACK [21] routine RSG on 50 pencils $(T,\ S)$ for different dimensions, where T is a symmetric tridiagonal random matrix with both diagonal and off-diagonal elements being uniformly distributed random numbers between 0 and 1. S is a symmetric tridiagonal matrix with off-diagonal elements, $s_{i,i+1}$, being uniformly distributed random numbers between 0 and 1, and its diagonal elements $s_{i,i} = 2\max\{s_{i-1,i}, s_{i,i+1}\}$.

The experiment was conducted on a Sun SPARC station IPC with IEEE standard and machine precision $\varepsilon \approx 2.2 \times 10^{-16}$.

For the symmetric tridiagonal generalized eigenproblem, the subroutine RSG in EISPACK is somewhat unattractive since it transforms the generalized eigenproblem $Tx = \lambda Sx$ to a standard eigenproblem $\tilde{T}y = \lambda y$ of a dense matrix $\tilde{T}$. It can not take advantage of the tridiagonal structure of the pencil (T,S) and clearly requires $O(n^3)$ flops. We conduct the comparison here mainly because it is the only algorithm available in EISPACK for eigenproblems of symmetric definite pencils.

Table 1 shows the results of our algorithm ZRST and the algorithm RSG. When eigenvectors are needed, inverse iteration is used in ZRST. So both algorithms can compute either all eigenvalues or all eigenpairs. The execution time apparently shows that the time complexities of ZRST and RSG are $O(n^2)$ and $O(n^3)$ respectively.

Order	Execution Time All eigenvalues		Execution Time All eigenpairs		$\max_i \frac{\|\|Tx_i - \lambda_i Sx_i\|\|_2}{\|\lambda_{max}\|}$		$\max_{i,j} \|(X^\top SX - I)_{i,j}\|$	
N	RGS	ZRST	RGS	ZRST	RGS	ZRST	RGS	ZRST
60	4.25	1.9	6.9	3.9	3.16×10^{-14}	8.32×10^{-15}	1.19×10^{-14}	4.91×10^{-14}
121	35.4	5.8	69.6	12.9	3.56×10^{-14}	1.75×10^{-14}	1.42×10^{-14}	1.63×10^{-14}
180	95.4	10.8	185.6	24.2	5.41×10^{-14}	2.83×10^{-15}	1.84×10^{-14}	8.02×10^{-14}
241	287.4	18.7	429.4	41.2	5.57×10^{-14}	7.10×10^{-14}	2.66×10^{-14}	5.73×10^{-14}

Table 1: Average execution time (second) of computed eigenvalues and eigenvectors.

Experiment 2. Let T be the Toeplitz matrix $[1,4,1]$ and S be the Toeplitz matrix

$[10^{-14}, 2\times 10^{-14}, 10^{-14}]$ with s_{11} and s_{nn} changed to 1. Although the condition number of S is about 10^{14}, the Crawford number $\sigma(T,S)$ is at least 2. So the eigenproblem of the pencil (T,S) is well-conditioned according to Theorem 4.5. The approximate eigenvalues of (T,S) evaluated by ZRST and RSG are denoted by $\{z_i\}_{i=1}^n$ and $\{r_i\}_{i=1}^n$ respectively.

To compare the accuracy of ZRST and RSG on (T,S), we use the eigenvalue counter $\kappa(\lambda)$ to conduct bisection and obtain intervals $\{[a_i,b_i]\}_{i=1}^n$ such that $\kappa(a_i)\le i-1$, $\kappa(b_i)\ge i$ and $|\arctan(b_i)-\arctan(a_i)|\le\varepsilon$. Notice that $\sigma=\sqrt{(2.51\varepsilon\|T\|_\infty)^2+(3.51\varepsilon\|S\|_\infty)^2}<16\varepsilon$. So the error bound in Theorem 4.5 is $\frac{\sigma}{c(T,S)}<8\varepsilon$. Let $[\hat a_i,\hat b_i]\supset[a_i,b_i]$ be the interval for which $|\arctan(\hat a_i)-\arctan(a_i)|=|\arctan(\hat b_i)-\arctan(b_i)|=8\varepsilon$. Then, by Corollary 4.2 and Theorem 4.5, the i^{th} smallest exact eigenvalue λ_i of the pencil (T,S) is in $[\hat a_i,\hat b_i]$. Let $\tilde\lambda_i=(a_i+b_i)/2$. Then

$$\begin{aligned}|\arctan(\tilde\lambda_i)-\arctan(\lambda_i)| &\le |\arctan(a_i)-\arctan(b_i)|+|\arctan(\hat a_i)-\arctan(a_i)|\\ &\le 9\varepsilon<2\times 10^{-15}.\end{aligned}$$

So if x is considered an approximate value of λ_i and $|x-\tilde\lambda_i|=\delta$, then

$$\delta-2\times10^{-15}\le|x-\lambda_i|\le\delta+2\times10^{-15}.$$

Let z_i's and r_i's be numerical eigenvalues obtained by ZRST and RSG respectively, Table 2 lists errors

$$\max_{1\le i\le n}\{|\arctan(z_i)-\arctan(\lambda_i)|\}\quad\text{and}\quad\max_{1\le i\le n}\{|\arctan(r_i)-\arctan(\lambda_i)|\}$$

for different matrix size n. The data clearly indicate that our algorithm is very accurate regardless of the condition number of matrix S, while RSG fails to compute some of the eigenvalues when S is ill-conditioned.

Order	error of ZRST	error of RSG
n	$\max_{1\le i\le n}\|\arctan(z_i)-\arctan(\lambda_i)\|$	$\max_{1\le i\le n}\|\arctan(r_i)-\arctan(\lambda_i)\|$
5	$\le 2.3\times10^{-15}$	$\le 2.2\times10^{-15}$
10	$\le 2.7\times10^{-15}$	$\ge 2.3\times10^{-7}$
20	$\le 2.5\times10^{-15}$	$\ge 2.5\times10^{-4}$
50	$\le 2.7\times10^{-15}$	≥ 1.07

Table 2: Accuracy comparison between ZRST and RSG.

Experiment 3. Bisection/Multisection method can also be used for the eigenproblem of symmetric tridiagonal pencils. However, we are unable to locate a suitable code in standard software packages specifically designed for our problem in the most general form. So instead, we tested the standard eigenvalue problem (i.e. $S=I$ and $T=[1,2,1]$ as a special case. Table 3 shows the computational result comparing our algorithm ZRST with bisection algorithm DSTEBZ in LAPACK [1] on this problem. Our algorithm ZRST is about three times faster than DSTEBZ.

For the special case $S=I$, the standard symmetric tridiagonal eigenproblem, more detailed results are reported in [14].

Matrix	Order	Execution Time (sec.)		Ratio	$\frac{\lvert\sum_{i=1}^{n}(t_{ii}-\lambda(i))\rvert}{\lvert\lambda\rvert_{max}}$	
Type	N	ZRST	DSTEBZ	DSTEBZ/ZRST	ZRST	DSTEBZ
$[1,2,1]$	65	0.76	2.42	3.18	1.22×10^{-15}	5.32×10^{-15}
	125	2.81	8.57	3.05	3.22×10^{-15}	1.06×10^{-14}
	255	11.35	34.78	3.06	8.66×10^{-15}	2.13×10^{-14}
	499	42.50	130.21	3.06	3.88×10^{-15}	7.99×10^{-15}

Table 3: The results of comparison of ZRST with DSTEBZ.

Experiment 4. Matrices T and S are obtained from piecewise linear finite element [22] discretization of the Sturm-Liouville problem

$$-\frac{d}{dx}\left(p(x)\frac{du}{dx}\right)+q(x)u=\lambda u,$$

where $u=u(x)$, $0<x<\pi$ and $u(0)=u'(\pi)=0$ and $p(x)>0$. When $[0,\pi]$ is divided into $n+1$ subintervals of equal length, the eigenvalue problem of the pencil (T,S) of order n is obtained. Here, both T and S are symmetric tridiagonal and positive definite. We use $p(x)=1$ and $q(x)=6$.

The computations were executed on BUTTERFLY GP 1000, a shared memory multiprocessor machine. The parallelization is naturally achieved by spreading Laguerre's iterations from different initial points to processors.

The *speed-up* is defined as

$$S_p=\frac{\text{execution time using one processor}}{\text{execution time using } p \text{ processors}}$$

and the *efficiency* is the ratio of the speed-up over p.

Table 4 shows the execution time and the speed-up S_p as well as the efficiency S_p/p of our algorithm.

Order	n =500					n =1000					
Nodes	1	4	8	16	32	1	4	8	16	32	64
(ExeTime)	368.4	96.64	53.48	30.99	18.75	1584.	407.9	217.7	120.7	70.76	45.78
S_p	1.0	3.81	6.89	11.9	19.7	1.0	3.88	7.28	13.1	22.4	34.6
S_p/p	1.0	0.95	0.86	0.74	0.61	1.0	0.97	0.91	0.82	0.70	0.54

Table 4: Execution time (second), speed-up and efficiency of the algorithm ZRST

8 Conclusions

We have described a method for computing generalized eigenvalues of symmetric tridiagonal definite pencil (T,S) of order n. Our method is apparantly more efficient than traditional algorithms in standard softwares even in serial mode. Our algorithm is naturally parallel in the sense that at each stage of the split-merge, there are n independent Laguerre's iterations which can be distributed to all available processors and each processor can operate with virtually no inter-processor communication except inputing an initial iteration point and outputing an eigenvalue. On the other hand, our algorithm is backward stable and the forward accuracy only depends on the Crawford number of the pencil, even when the matrix S is extremely ill-conditioned, for which the traditional approach may fail to compute eigenvalues in meaningful accuracy.

References

[1] E. Anderson, et.al., *LAPACK User's Guide,* SIAM, Philadelphia, 1992.

[2] A. E. Berger, J. M. Solomon and M. Ciment, *Higher order accurate tridiagonal difference methods for diffusion convection equations*, in *Advances in Computer Methods for Partial Differential Equations III*, R. Vichnevetsky and R. S. Stepleman, editors, pp 322–330, IMACS, 1978.

[3] C. F. Borges and W. B. Gragg, *A parallel divide and conquer algorithm for the generalized real symmetric definite tridiagonal eigenproblem*, preprint, Naval Postgraduate School, 1992

[4] R. F. Boisvert, *Families of higher order accurate discretizations of some elliptic problems*, SIAM J. Sci. Stat. Comput., vol. 2, pp 268–284, 1981.

[5] L. Collatz, *Numerical Treatment of Differential Equations*, Springer, Berlin, 1960.

[6] J. J. M Cuppen, *A Divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math., vol. 36, pp 177–195, 1981.

[7] J. J. Dongarra and D. C. Sorensen, *A fully parallel Algorithm for the symmetric eigenvalue problem*, SIAM J. Sci. Stat. Comput., vol. 8, pp 139–154, 1987.

[8] G. H. Golub and Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD (1989).

[9] B. Grieves and D. Dunn, *Symmetric matrix methods for Schrödinger eigenvectors*, J. Phys. A: Math Gen., vol. 23, pp 5479–5491, 1990.

[10] W. Kahan, *Notes on Laguerre's iteration*, unpublished research notes, 1992.

[11] L. Kaufman, *An algorithm for the banded symmetric generalized matrix eigenvalue problem*, SIAM J. Matrix Anal. Appl., vol. 14, No. 2, pp 372-389, 1993

[12] K. Li and T. Y. Li, *An Algorithm For Symmetric Tridiagonal Eigen-problems — Divide and Conquer with Homotopy Continuation,* SIAM J. Sci. Comput. vol 14, No. 3, pp 735–751, 1993.

[13] K. Li and T. Y. Li, *A homotopy algorithm for a symmetric generalized eigenproblem*, Numerical Algorithms. vol 4, pp 167–195, 1993.

[14] T. Y. Li and Z. Zeng, *Laguerre's Iteration in Solving The Symmetric Tridiagonal Eigenproblem — Revisited,* To appear: SIAM J. Sci. Comput.

[15] S-S. Lo, B. Phillipe and A. Sameh, *A multiprocessor algorithm for the symmetric tridiagonal eigenvalue problems*, SIAM J. Sci. Stat. Comput., vol. 8, pp 155–165, 1987.

[16] R. D. Pantazis and D. B. Szyld, *A multiprocessor method for the solution of the generalized eigenvalue problem on an interval*, Parallel Processing for Scientific Computing, Proceedings of the Fourth SIAM Conference, Jack J. Dongarra and Paul Messina and Danny G. Sorensen and Robert G. Voigt, ed. SIAM Publications, Philadelphia, PA, pp 36–41, 1990.

[17] B. N. Parlett, *The symmetric eigenvalue problem,* Prentice-Hall, inc., Englenwood, Cliffs, NJ, 1980.

[18] B. Philippe and B. Vital, *Parallel Implementations for Solving Generalized Eigenvalue Problems with Symmetric Sparse Matrices*, Applied Numerical Mathematics, vol. 12, pp 391–402, 1993.

[19] Y. M. Ram and G. M. L. Gladwell, *Constructing a finite-element model of a vibratory rod from eigendata*, J. of Sound and Vibration, vol. 165, No. 1, to appear, 1993.

[20] C. J. Ribbens and C. Beattie, *Parallel solution of generalized symmetric tridiagonal eigenvalue problem on shared memory multiprocessors*, Technical Report TR 92-47, Department of Computer Science, Virginia Polytechnic Institute and State University, 1992.

[21] B. T. Smith et al., *Matrix Eigensystem Routines — EISPACK Guide*, 2nd ed., Springer, New York 1976.

[22] G. Strang and G. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1973.

[23] C. Trefftz, P. McKinley, T. Y. Li and Z. Zeng, *A scalable eigenvalue solver for symmetric tridiagonal matrices*, Proceedings of The Sixth SIAM Conference on Parallel Processing for Scientific Computing, SIAM publications, pp 602–609, 1993.

[24] W. Weaver, Jr. and P. R. Johnson, *Finite Elements for structural Analysis*, Prentice-Hall, New Jersey, 1984

[25] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, 1965.