The Scaled Sturm Sequence Computation

J. Zhang*

1 Introduction

The Sturm sequence computation is used by the bisection method to compute eigenvalues of real symmetric tridiagonal matrices. Let T_n be a symmetric tridiagonal matrix with the diagonal elements $\alpha_1, \alpha_2, \ldots, \alpha_n$ and the off-diagonal elements $\beta_1, \beta_2, \ldots, \beta_{n-1}$. Given a number λ, the sequence of characteristic polynomials $p_j(\lambda)$ for the leading $j \times j$ principal submatrices of T_n can be computed with the following three-term linear recurrence [7, 4].

$$\begin{aligned} &p_0(\lambda) = 1, \\ &p_1(\lambda) = \alpha_1 - \lambda, \\ &p_j(\lambda) = (\alpha_j - \lambda)p_{j-1}(\lambda) - \beta_{j-1}^2 p_{j-2}(\lambda), \quad j = 2, \cdots, n. \end{aligned} \tag{1}$$

The sequence $\{p_j(\lambda)\}$ is referred to as the *Sturm sequence*. In this paper, it is called the *classical Sturm sequence*. It is well known[4] that the number of eigenvalues smaller than λ is equal to the number of sign disagreements between consecutive members in the classical Sturm sequence. Therefore, given that the ith eigenvalue of T_n is located in an interval, we can extract the ith eigenvalue to meet the desired precision by repeated bisection of the interval based on the classical Sturm sequence evaluation. Due to its overflow and underflow problems, in real application the classical Sturm sequence computation gives way to the following variant [1]

$$\begin{aligned} &q_1(\lambda) = \alpha_1 - \lambda, \\ &q_j(\lambda) = \alpha_j - \lambda - \beta_{j-1}^2 / q_{j-1}(\lambda), \quad j = 2, \cdots, n. \end{aligned} \tag{2}$$

where $q_j(\lambda) = p_j(\lambda)/p_{j-1}(\lambda)$. The sequence $\{q_j(\lambda)\}$ is called the *Sturm sequence* in this paper. With the Sturm sequence, the number of eigenvalues that are less

*Department of Computer Science University of Illinois at Springfield, Springfield, IL 62703

than λ is equal to the number of negative members in the sequence. Because of its self scaling, the Sturm sequence computation (2) can avoid overflow and underflow problems. It can be noted that the Sturm sequence computation may break down if $q_j(\lambda) = 0$ for some $1 \leq j < n$. To prevent this, the following guardian [1] is designed: if $q_{j-1}(\lambda) = 0$, set $q_j(\lambda) = \alpha_j - \lambda - |\beta_{j-1}|/\epsilon$ where ϵ is the machine epsilon. Applying the guardian is equivalent to introducing a perturbation $|\beta_{j-1} \cdot \epsilon|$ to the diagonal element α_{j-1}.

The bisection with the Sturm sequence computation (2) is quite accurate in computing eigenvalues of T_n. However, due to the nonlinear nature of the recurrence used for the computation, the Sturm sequence is difficult to be parallelized. In this paper, the *scaled Sturm sequence computation* is presented by modifying the classical Sturm sequence computation. The scaled Sturm sequence computation is suitable for being parallelized. It is shown that the scaled Sturm sequence computation is backward stable and is capable of avoiding overflow and underflow problems. The numerical result shows that the scaled Sturm sequence computation achieves the same accuracy in computing eigenvalues of T_n as that of the Sturm sequence computation although its running time is about one and a half times of that taken by the Sturm sequence computation.

To compute eigenvalues of T_n correctly, it is important to avoid or handle the *nonmonotonicity* for the result of the Sturm sequence computation [3]. The nonmonotonicity denotes the phenomenon that the number of negative members of the Sturm sequence computed for one number x is less than that of another number $y < x$. The scaled Sturm sequence computation can be another resort in case the nonmonotonicity occur in the Sturm sequence computation.

2 The scaled Sturm sequence computation

The scaled Sturm sequence computation is developed based on the classical sturm sequence computation. As observed in [2], the classical Sturm sequence computation can be represented with the matrix-vector multiplication

$$P_i \;=\; M_i P_{i-1} \tag{3}$$

where

$$M_i \;=\; \begin{bmatrix} \alpha_i - \lambda & -\beta_{i-1}^2 \\ 1 & 0 \end{bmatrix}, \quad P_i \;=\; \begin{bmatrix} p_i(\lambda) \\ p_{i-1}(\lambda) \end{bmatrix}. \tag{4}$$

Note that P_0 is defined to be $[1,\ 0]^T$ and β_0 is set to 0. The first elements $P_i[1]$ of those vectors P_i form the classical Sturm sequence. To prevent the overflow and underflow problems, we use a scaling factor to multiply with P_i when necessary. Therefore, the scaled Sturm sequence computation can be expressed as the following sequence of matrix-vector multiplications.

$$P_i = s_i M_i P_{i-1} (i = 1, 2, \ldots, n) \tag{5}$$

where s_i is the scaling factor used in step i, which is computed with the following algorithm.

The Scaling factor algorithm
Input: $\widehat{P}_i = M_i P_{i-1}$
output: The scaling factor s_i
$w = \max\{|\widehat{P}_i[1]|, |\widehat{P}_i[2]|\}$
if $w > \Phi$ then $s_i = \Phi/w$
else if $w < \Upsilon$ then $s_i = \Upsilon/w$ else $s_i = 1$
return s_i

The values of Φ and Υ are set to 10^{10} and 10^{-10} respectively in our implementation. The sequence $\{P_i[1]\}$ is referred to as the scaled Sturm sequence. It is easy to see that due to the scaling operations, the first and second elements of P_i are equal to $C \cdot p_i(\lambda)$ and $C \cdot p_{i-1}(\lambda)$ respectively where $p_i(\lambda)$ is the ith member of the classical Sturm sequence and C is a positive constant. Therefore, the count of sign disagreements for the scaled Sturm sequence $\{P_i[1]\}$ is the same as that of the classical Sturm sequence $\{p_i(\lambda)\}$. It should be mentioned that in the real implementation, the scaled Sturm sequence is not calculated with the matrix-vector multiplication. Instead, a more efficiently designed algorithm is used to compute the members of the scaled Sturm sequence and count the number of sign disagreements. We put the scaled Sturm sequence computation in this form just for the clearance and convenience in the analysis below. In the following discussion, we call the scaled Sturm sequence computation (5) the vector form of the scaled Sturm sequence computation. In showing the backward stability of the scaled Sturm sequence computation, we need the following remark.

Remark 2.1: In the matrix-vector multiplication $\widehat{P}_i = M_i P_{i-1}$, since the second row of M_i is $[1,\ 0]$, $\widehat{P}_i[2]$ is always equal to $P_{i-1}[1]$. To simulate the real algorithm for the scaled Sturm sequence computation, we consider that the value of the element $\widehat{P}_i[2]$ is directly set to the value of the element $P_{i-1}[1]$ instead of being computed as the inner product of the vectors $[0,\ 1]^T$ and P_{i-1}.

As to the possibility of being parallelized, it can be seen from (5) that $P_i = s_i M_i s_{i-1} M_{i-1} \ldots s_1 M_1 P_0$. Therefore, the computation of P_i $(1 \le i \le n)$ can be parallelized by the typical prefix computation of

$$P_n = s_n M_n s_{n-1} M_{n-1} \ldots s_1 M_1 P_0.$$

Such a parallel algorithm for computing the scaled Sturm sequence is presented in [8, 9].

As an example similar to the one in [3], the following matrix will incur the nonmonotonicity for the Sturm sequence computation at the points -10^{-32} and 0 when the guardian described in section 1 is used to deal with the zero denominator.

$$A = \begin{bmatrix} 0 & 2\epsilon \\ 2\epsilon & 3 \end{bmatrix}$$

Here, $\epsilon = 1.11 \times 10^{-16}$. It is easy to verify that the scaled Sturm sequence computation can avoid the nonmonotonicity in the example. It should be mentioned

that some more sophisticated guardian for the Sturm sequence computation like the one in [6] can avoid the nonmonotonicity with the above example. In spite of this, the scaled Sturm sequence computation can be one more resort to deal with the nonmonotonicity.

3 The backward stability

The following theorem shows that the scaled Sturm sequence computation is backward stable.

Theorem 3.1: Let $P_i = s_i M_i P_{i-1}$ $(i = 1, 2, \ldots, n)$ be the vector form of the scaled Sturm sequence computation for the symmetric tridiagonal matrix T_n with the diagonal elements α_i $(i = 1, 2, \ldots, n)$ and the off-diagonal elements β_i $(i = 1, 2, \ldots, n-1)$ at the number λ where $P_0 = (1, 0)^T$. Then, the first component $P_i[1]$ of each P_i is the exact result computed from the symmetric tridiagonal matrix $\widetilde{T}_n$ with the diagonal elements $\widetilde{\alpha}_i$ $(i = 1, 2, \ldots, n)$ and the off-diagonal elements $\widetilde{\beta}_i$ $(i = 1, 2, \ldots, n-1)$ where $\widetilde{\alpha}_i$ and $\widetilde{\beta}_i$ are obtained by introducing perturbations $\Delta\alpha_i$ and $\Delta\beta_i$ to α_i and β_i respectively. Furthermore, $|\Delta\alpha_i| \leq 4\epsilon(\max_{1\leq k\leq n}\{|\alpha_k|\} + |\lambda|)$ for $i = 1, 2, \ldots, n$ and $|\Delta\beta_i| < 2.5\epsilon \max_{1\leq k\leq n-1}\{|\beta_k|\}$ for $i = 1, 2, \ldots, n-1$ where ϵ is the machine epsilon.

Proof: The result can be proved for each $P_i[1]$ $(i = 1, 2, \ldots, n)$ by induction on i. For the base case, it can be seen that $P_1[1]$ is computed as $P_1[1] = \alpha_1 \ominus \lambda$ where $\ominus$ denotes the numerical minus operation. With the fundamental axiom of the floating point arithmetic, $P_1[1] = (\alpha_1 - \lambda)(1 + \epsilon) = \alpha_1 + \epsilon(\alpha_1 - \lambda) - \lambda$. Therefore, $P_1[1]$ is the exact result computed from $\widetilde{T}_n$ with $\widetilde{\alpha}_1 = \alpha_1 + \Delta\alpha_1$ where $\Delta\alpha_1 = \epsilon(\alpha_1 - \lambda)$. It can be seen that $|\Delta\alpha_1| = \epsilon|\alpha_1 - \lambda| \leq \epsilon(|\alpha_1| + |\lambda|) < 4\epsilon(\max_{1\leq k\leq n}\{|\alpha_k|\} + |\lambda|)$. Therefore, the result holds for the base case.

For the induction hypothesis, assume that for $2 \leq j \leq i$, the result of the theorem holds for $P_{j-1}[1]$. From the computation $P_i = s_i M_i P_{i-1}$,

$$P_i[1] = s_i \otimes [(\alpha_i \ominus \lambda) \otimes P_{i-1}[1] \ominus (\beta_{i-1} \otimes \beta_{i-1}) \otimes P_{i-1}[2]] \tag{6}$$

where like $\ominus$, $\otimes$ represents the numerical multiplication operation. For the item $P_{i-1}[2]$ in (6), consider the computation $P_{i-1} = s_{i-1} M_{i-1} P_{i-2}$. According to Remark 2.1, we directly use $P_{i-2}[1]$ as the second element of the resulting vector $M_{i-1}P_{i-2}$. Therefore, $P_{i-1}[2] = s_{i-1} \otimes P_{i-2}[1]$. Plugging the result into (6), we have

$$P_i[1] = s_i \otimes [(\alpha_i \ominus \lambda) \otimes P_{i-1}[1] \ominus (\beta_{i-1} \otimes \beta_{i-1}) \otimes (s_{i-1} \otimes P_{i-2}[1])] \tag{7}$$

Repeatedly applying the fundamental axiom of the floating point arithmetic, we have

$$P_i[1] = s_i[(\alpha_i - \lambda)P_{i-1}[1](1 + \epsilon_1)(1 + \epsilon_2)(1 + \epsilon_6)(1 + \epsilon_7) - \beta_{i-1}^2 s_{i-1} P_{i-2}[1]$$

$$(1+\epsilon_3)(1+\epsilon_4)(1+\epsilon_5)(1+\epsilon_6)(1+\epsilon_7)] \tag{8}$$

For the item $(1+\epsilon_1)(1+\epsilon_2)(1+\epsilon_6)(1+\epsilon_7)$ in (8), unfolding the expression and eliminating the items of higher order, we have $(1+\epsilon_1)(1+\epsilon_2)(1+\epsilon_6)(1+\epsilon_7) \approx 1+\epsilon_1+\epsilon_2+\epsilon_6+\epsilon_7 \approx 1+4\epsilon$. Similarly, for the item $(1+\epsilon_3)(1+\epsilon_4)(1+\epsilon_5)(1+\epsilon_6)(1+\epsilon_7)$ in (8), we can approximate it with $1+5\epsilon$. Plugging the results into (8), we have

$$P_i[1] = s_i[(1+4\epsilon)(\alpha_i-\lambda)P_{i-1}[1]) - (1+5\epsilon)\beta_{i-1}^2 s_{i-1}P_{i-2}[1]]. \tag{9}$$

For the item $(1+4\epsilon)(\alpha_i-\lambda)$ in (9), it is easy to see that it can be replaced with $\alpha_i+\Delta\alpha_i-\lambda$ with $\Delta\alpha_i = 4\epsilon(\alpha_i-\lambda)$. For $\Delta\alpha_i$, we have $|\Delta\alpha_i| = 4\epsilon|\alpha_i-\lambda| \leq 4\epsilon(|\alpha_i|+|\lambda|) \leq 4\epsilon(\max_{1\leq k\leq n}\{|\alpha_k|\}+|\lambda|)$.

For the item $(1+5\epsilon)\beta_{i-1}^2$ in (9), in order to express it with $(\beta_{i-1}+\Delta\beta_{i-1})^2$, $\Delta\beta_{i-1}$ should satisfy the equation

$$(\beta_{i-1}+\Delta\beta_{i-1})^2 = (1+5\epsilon)\beta_{i-1}^2 \tag{10}$$

Since T_n is an unreduced symmetric tridiagonal matrix, $|\beta_{i-1}|$ is great enough to meet the criterion for deflating small off-diagonal elements. Therefore, it is reasonable to assume that the perturbation $\Delta\beta_{i-1}$ will not change the sign of β_{i-1}, that is, $\beta_{i-1}+\Delta\beta_{i-1}$ and β_{i-1} should have the same sign. Then, from (10), it follows that

$$\beta_{i-1}+\Delta\beta_{i-1} = \sqrt{1+5\epsilon}\beta_{i-1} \tag{11}$$

From (11), we have

$$\begin{aligned}\Delta\beta_{i-1} &= (\sqrt{1+5\epsilon}-1)\beta_{i-1} \\ &= \frac{5\epsilon}{1+\sqrt{1+5\epsilon}}\beta_{i-1}\end{aligned} \tag{12}$$

From (12), it follows that $|\Delta\beta_{i-1}| < 2.5\epsilon\cdot|\beta_{i-1}| \leq 2.5\max_{1\leq k\leq n-1}\{|\beta_k|\}\epsilon$. Therefore, by the induction hypothesis, the result of the theorem also holds for P_i. Hence, we completed the induction.□

4 The effectiveness of eliminating the overflow and underflow problems

In order to show that the scaled Sturm sequence computation can eliminate the overflow and underflow problems, we need to bound the absolute values of elements in a symmetric tridiagonal matrix T_n with the diagonal elements α_i $(i=1,2,\ldots,n)$ and the off-diagonal elements β_i $(i=1,2,\ldots,n-1)$. The absolute values of those elements satisfy the following constraints.

$$\begin{aligned}|\alpha_i| &\leq \phi \quad (1\leq i\leq n) \\ \upsilon &\leq |\beta_i| \leq \phi \quad (1\leq i\leq n-1).\end{aligned} \tag{13}$$

When some elements of T_n are greater than ϕ in absolute value, we can scale T_n to make the absolute values of elements of T_n bounded by ϕ. The lower bound υ can be appropriately set to meet the criterion for deflating small off-diagonal elements. Therefore, the off-diagonal elements that are less than υ in absolute value can be deflated to 0, and the eigenproblem of T_n will be decomposed into eigenproblems of those resulting unreduced symmetric tridiagonal submatrices. In the following discussion, we select 10^{-15} as the value of υ since it is near the machine precision ϵ. For ϕ, we select the value 10^{15}.

The Gershgorin Circle Theorem [4] is used in the proof of some theorem and proposition listed below. So, it is proper to state the theorem here. According to the Gershgorin Circle Theorem, all the eigenvalues of T_n are contained in the interval

$$[\min_{1\le i\le n}\{\alpha_i - (|\beta_{i-1}| + |\beta_i|)\}, \quad \max_{1\le i\le n}\{\alpha_i + (|\beta_{i-1}| + |\beta_i|)\}]. \tag{14}$$

In practice, interval (14) is used as the initial interval to find eigenvalues of T_n with the bisection method.

The following theorem shows that there is no overflow problem in the scaled Sturm sequence computation.

Theorem 4.1: There is no overflow problem for the computation P_i in the vector form of the scaled Sturm sequence computation (5).

From the element value restriction (13), the scaling operation, and the Gershgorin Circle Theorem (14), it is not difficult to derive that the absolute value of elements in P_i is bounded by $2\Phi \cdot \phi^2$ in the scaled Sturm sequence computation. With the values set for Φ and ϕ, $2\Phi \cdot \phi^2 = 2 \times 10^{40}$ that is far below the limit of large real numbers an ordinary machine expresses. For instance, in Sun Sparc workstations, the maximum real number expressed with the double precision has the same order of magnitude as that of 10^{324}. The proof of Theorem 4.1 can be found in [9].

The following two propositions are needed to show that the scaled Sturm sequence computation is valid in eliminating the underflow problem.

Proposition 4.1: In the vector form of the scaled Sturm sequence computation (5), $P_i \neq [0,\ 0]^T$ for $i = 1, 2, \ldots, n$.

Proof: The proposition can be proved by induction on i. For the base case, when $i = 1$, from the configuration of M_i, $P_1[2] = P_0[1]$. Therefore, by $P_0[1] = 1$, P_1 is a nonzero vector.

For the induction hypothesis, assume that $P_{i-1} \neq [0,\ 0]^T$ for $i > 1$. Let

$$\widehat{P}_i = M_i P_{i-1} = \begin{bmatrix} \alpha_i - \lambda & -\beta_{i-1}^2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} P_{i-1}[1] \\ P_{i-1}[2] \end{bmatrix}. \tag{15}$$

In the following, we show that $\widehat{P}_i$ is a nonzero vector. When $P_{i-1}[1] \neq 0$, from

(15), $\widehat{P}_i[2] = P_{i-1}[1]$. Therefore, $\widehat{P}_i \neq [0,\ 0]^T$. Otherwise, when $P_{i-1}[1] = 0$, from (15), we have

$$\widehat{P}_i[1] = -\beta_{i-1}^2 P_{i-1}[2]. \tag{16}$$

By the induction hypothesis, $P_{i-1} \neq [0,\ 0]^T$. Therefore, by $P_{i-1}[1] = 0$ and the scaling operation, $|P_{i-1}[2]| \geq \Upsilon$. Applying the result and the element value restriction (13) to (16), we have $|\widehat{P}_i[1]| \geq \Upsilon \cdot \upsilon^2$. With the values set to Υ and υ, $\Upsilon \cdot \upsilon^2 = 10^{-40}$ which is far above the limit of small positive real numbers an ordinary machine can express. For instance, the minimum positive real number expressed with double precision in Sun Sparc workstations is equal to 10^{-324} in the order of magnitude. Therefore, no underflow will occur in the computation of $\widehat{P}_i[1]$ and $\widehat{P}_i \neq [0,\ 0]^T$. Then, by the scaling operation, $P_i \neq [0,\ 0]^T$. Hence, induction is completed.□

In the following proposition and theorem, p_j is the characteristic polynomial of the leading $j \times j$ principal submatrix of T_n.

Proposition 4.2: When λ is not equal to any zero of p_j and p_{j-1} $(2 \leq j \leq n)$, then $|p_j(\lambda)|/|p_{j-1}(\lambda)| > |\lambda_{k,j} - \lambda|$ and $|p_{j-1}(\lambda)|/|p_j(\lambda)| > \frac{1}{9}\phi^{-2}|\lambda_{k',j-1} - \lambda|$ where $\lambda_{k,j}$ is the zero of p_j which is the closest to λ and $\lambda_{k',j-1}$ is the zero of p_{j-1} which is the closest to λ.

It is not difficult to prove the proposition based on the property that the zeros of p_j and p_{j-1} interlace with each other [7] as well as the Gershgorin Circle Theorem (14). The proof of the proposition can be found in [9].

The following theorem shows the correctness of the scaled Sturm sequence computation in eliminating the underflow problem.

Theorem 4.2: For one step of the vector form of the scaled Sturm sequence (5), there will be no underflow problem in the computation of $P_i[1]$ if λ is not close to any zero λ_j of p_i with respect to the following criterion (17).

$$|\lambda - \lambda_j| > \epsilon^2 \tag{17}$$

There will be no underflow problem in the computation of $P_i[2]$ if λ is not close to any zero λ_j of p_{i-1} with respect to the criterion (17).

Proof: Let $\widehat{P}_i = M_i P_{i-1}$. Then,

$$\begin{bmatrix} \widehat{P}_i[1] \\ \widehat{P}_i[2] \end{bmatrix} = \begin{bmatrix} \alpha_i - \lambda & -\beta_{i-1}^2 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_{i-1}[1] \\ P_{i-1}[2] \end{bmatrix}. \tag{18}$$

Assume that λ is not close to any zero of p_i with respect to the criterion (17). In the following, we show that there is no underflow problem in the computation of $\widehat{P}_i[1]$ in (18).

By Proposition 4.1, P_{i-1} is a nonzero vector. When $|P_{i-1}[1]| \geq |P_{i-1}[2]|$, due to the scaling operation, $|P_{i-1}[1]| \geq \Upsilon$. From (18), $\widehat{P}_i[2] = P_{i-1}[1]$. Thus,

$$|\widehat{P}_i[2]| \geq \Upsilon. \tag{19}$$

As we know, $P_{i-1}[1] = Cp_{i-1}$ and $P_{i-1}[2] = Cp_{i-2}$ where C is a positive constant. Therefore, by Proposition 4.2,

$$\frac{|\widehat{P}_i[1]|}{|\widehat{P}_i[2]|} = \frac{|Cp_i(\lambda)|}{|Cp_{i-1}(\lambda)|} = \frac{|p_i(\lambda)|}{|p_{i-1}(\lambda)|} > |\lambda_{k,i} - \lambda|. \tag{20}$$

where $\lambda_{k,i}$ is the zero of p_i that is the closest to λ. According to (19) and (20), $|\widehat{P}_i[1]| > \Upsilon \cdot |\lambda_{k,i} - \lambda|$. Since λ is not close to any zero of p_i with respect to the criterion (17), we have $|\widehat{P}_i[1]| > \Upsilon \cdot \epsilon^2$. With the values set for Υ and ϵ, $|\widehat{P}_i[1]| > 10^{-42}$ that is far above the limit of small positive real numbers an ordinary machine can express.

When $|P_{i-1}[1]| < |P_{i-1}[2]|$, we will discuss the situation according to the following two cases.
(i) $|P_{i-1}[1]| < \frac{1}{8}\upsilon^2 \cdot \Upsilon \cdot \phi^{-1}$

From (18), $\widehat{P}_i[1] = (\alpha_i - \lambda)P_{i-1}[1] - \beta_{i-1}^2 \cdot P_{i-1}[2]$. Since $|P_{i-1}[1]| < |P_{i-1}[2]|$, due to the scaling operation, $|P_{i-1}[2]| \geq \Upsilon$. By the element value restriction (13), $\beta_{i-1}^2 \geq \upsilon^2$. Therefore, $\beta_{i-1}^2|P_{i-1}[2]| \geq \upsilon^2\Upsilon$. According to the element value restriction (13) and the Gershgorin Circle Theorem (14), $|\alpha_i - \lambda| \leq 4\phi$. Therefore, by $|P_{i-1}[1]| < \frac{1}{8}\upsilon^2 \cdot \Upsilon \cdot \phi^{-1}$, $|\alpha_i - \lambda||P_{i-1}[1]| < \frac{1}{2}\upsilon^2 \cdot \Upsilon$. Therefore, $|\widehat{P}_i[1]| = |(\alpha_i - \lambda)P_{i-1}[1] - \beta_{i-1}^2 \cdot P_{i-1}[2]| > \frac{1}{2}\upsilon^2 \cdot \Upsilon$. With the values set for υ and Υ, there is no underflow in the computation of $\widehat{P}_i[1]$ in this case.
(ii) $|P_{i-1}[1]| \geq \frac{1}{8}\upsilon^2 \cdot \Upsilon \cdot \phi^{-1}$

From (18), $\widehat{P}_i[2] = P_{i-1}[1]$. Thus, $|\widehat{P}_i[2]| \geq \frac{1}{8}\upsilon^2 \cdot \Upsilon \cdot \phi^{-1}$. Therefore, by Proposition 4.2, $|\widehat{P}_i[1]| > |\lambda_{k,i} - \lambda||\widehat{P}_i[2]| \geq \frac{1}{8}\upsilon^2 \cdot \Upsilon \cdot \phi^{-1}|\lambda_{k,i} - \lambda|$ where $\lambda_{k,i}$ is the zero of p_i that is the closest to λ. Since λ is not close to any zero of p_i with respect to the criterion (17), from the above result, $|\widehat{P}_i[1]| > \frac{1}{8}\upsilon^2 \cdot \Upsilon \cdot \phi^{-1} \cdot \epsilon^2 > 10^{-88}$ with the values set for υ, Υ, ϕ, and ϵ. Therefore, there is still no underflow in the computation of $\widehat{P}_i[1]$ in this case.

In the above, we showed that if λ is not close to any zero of p_i with respect to the criterion (17), then there is no underflow in the computation of $\widehat{P}_i[1]$ in (18). Therefore, when s_i in (5) is greater than 1, $P_i[1] = s_i\widehat{P}_i[1]$ will have no underflow in its computation. When $s_i < 1$, by Proposition 4.1, P_i is a nonzero vector. With the scaling operation, it can be seen that $\max\{|P_i[1]|, |P_i[2]|\} \geq \Upsilon$. Therefore, if $|P_i[1]| = \max\{|P_i[1]|, |P_i[2]|\}$, there is naturally no underflow in the computation of $P_i[1]$ in (5). When $|P_i[2]| = \max\{|P_i[1]|, |P_i[2]|\}$, by Proposition 4.2, $|P_i[1]| > |\lambda_{k,i} - \lambda||P_i[2]| \geq \Upsilon|\lambda_{k,i} - \lambda|$. By the assumption that λ is not close to any zero of p_i with respect to the criterion (17), $|P_i[1]| > \epsilon^2 \cdot \Upsilon \geq 10^{-42}$. Therefore, there is no underflow in the computation of $P_i[1]$ in (5).

We now prove the second part of the theorem. Assume that λ is not close to any zero of p_{i-1}. With the above result, there will be no underflow in the computation of $P_{i-1}[1]$ in $P_{i-1} = s_{i-1}M_{i-1}P_{i-2}$. Let $\widehat{P}_i = M_iP_{i-1}$. From the configuration of M_i, $\widehat{P}_i[2] = P_{i-1}[1]$. Therefore, if the scaling factor s_i is greater

than 1, there will be naturally no underflow in the computation of $P_i[2]$. If $s_i < 1$, we will make the proof according to the following two cases.

When $|\widehat{P}_i[2]| \geq |\widehat{P}_i[1]|$, from the scaling operation, $P_i[2] = s_i\widehat{P}_i[2]$ will be equal to Φ in absolute value. Therefore, there is naturally no underflow in the computation of $P_i[2]$.

When $|\widehat{P}_i[2]| < |\widehat{P}_i[1]|$, by Proposition 4.2,

$$\frac{|P_i[2]|}{|P_i[1]|} = \frac{|\widehat{P}_i[2]|}{|\widehat{P}_i[1]|} = \frac{|p_{i-1}(\lambda)|}{|p_i(\lambda)|} > \frac{1}{9}\phi^{-2}|\lambda_{k,i-1} - \lambda| \tag{21}$$

where $\lambda_{k,i-1}$ is the zero of p_{i-1} that is the closest to λ. From the scaling operation, $P_i[1] = s_i\widehat{P}_i[1]$ will be equal to Φ in absolute value. Therefore, from (21), $|P_i[2]| > \frac{1}{9}\phi^{-2}|\lambda_{k,i-1} - \lambda|\Phi$. By the assumption that λ is not close to any zero of p_{i-1} with respect to the criterion (17), we have $|P_i[2]| > \frac{1}{9}\phi^{-2}\epsilon^2\Phi > 10^{-53}$ with the values set for ϕ, ϵ, and Φ. Therefore, there is still no underflow in the computation of $P_i[2]$ in this case.□

From the proof of Theorem 4.2, the threshold in the criterion (17) can be set to be much smaller than ϵ^2 and the theorem will still hold. When the distance between λ and a zero of p_i or p_{i-1} is much less than ϵ^2, underflow may occur in computing $P_i[1]$ or $P_i[2]$ in (5). In this case, we can consider that λ is an approximation of a zero of p_i or p_{i-1}. From the convention of counting sign disagreements between two consecutive members of the Sturm sequence [4], this type of underflows will not influence the correct count of the sign disagreements.

5 Numerical result

In the numerical experiment, the Sturm sequence computation and the scaled Sturm sequence computation are tested on the Sun workstation. Table 1 lists their accuracy and time results for the computation of particular eigenvalues of the following testing matrices presented in [5]. For those testing matrices, the error of the eigenvalue evaluation can be directly evaluated as

$$|\widetilde{\lambda}_i - \lambda_i|/\|T_n\|.$$

where $\widetilde{\lambda}_i$ is the approximation of the exact eigenvalue λ_i and $\|T_n\| = \|T_n\|_\infty = \max_{1\leq i\leq n}(|\beta_{i-1}| + |\alpha_i| + |\beta_i|)$. The machine epsilon is set as $\epsilon = 1.11 \times 10^{-16}$.

Type 1: Toeplitz matrices $[b, a, b]$ with $\alpha_i = a$ and $\beta_i = b$. Exact eigenvalues: $\{a + 2b\cos\frac{k\pi}{n+1}\}_{1\leq k\leq n}$. In generating testing matrices, a and b are set to 0.2 and 0.1 respectively.

Type 2: $\alpha_1 = a-b$, $\alpha_i = a$ for $i = 2, \ldots, n-1$, $\alpha_n = a+b$. $\beta_i = b$ for $i = 1, \ldots, n-1$. Exact eigenvalues: $\{a+2b\cos\frac{(2k-1)\pi}{2n}\}_{1\leq k\leq n}$. In generating testing matrices, a and b are set to 0.2 and 0.1 respectively.

Type 3: $\alpha_i = \begin{cases} a & \text{for odd } i, \\ b & \text{for even } i, \end{cases}$ and $\beta_i = 1$. Exact eigenvalues:

$$\left\{\frac{a+b\pm[(a-b)^2+16\cos^2\frac{k\pi}{n+1}]^{1/2}}{2}\right\}_{1\le k\le n/2} \quad \text{and } a \text{ if } n \text{ is odd.}$$

In generating testing matrices, a and b are set to 0.2 and 0.1 respectively.

Type 4: $\alpha_i = -[(2i-1)(n-1)-2(i-1)^2]$, $\beta_i = i(n-i)$. Exact eigenvalues: $\{-k(k-1)\}_{1\le k\le n}$.

Table 1. *The accuracy and time comparison between the Sturm sequence and the scaled Sturm sequence (The order of matrices is* 2000*; the time unit is the microsecond, i.e.,* 10^{-6} *sec.)*

matrix type	eigenvalue index	Sturm		Scaled Sturm	
		accuracy	time	accuracy	time
type 1	1	0.19ϵ	0.012	0.19ϵ	0.020
	1000	0	0.012	0	0.019
	2000	1.25ϵ	0.012	1.25ϵ	0.019
type 2	1	0.57ϵ	0.012	0.57ϵ	0.020
	1000	1.25ϵ	0.012	1.25ϵ	0.019
	2000	0	0.012	0	0.019
type 3	1	0	0.012	0	0.013
	1000	0.23ϵ	0.012	0.23ϵ	0.014
	2000	1.82ϵ	0.012	1.82ϵ	0.014
type 4	1	0	0.012	0	0.019
	1000	0.52ϵ	0.012	0.52ϵ	0.020
	2000	0.50ϵ	0.012	0.50ϵ	0.020

From table 1, the scaled Sturm sequence has the same accuracy result as that of the Sturm sequence. In fact, the bisection with the scaled Sturm sequence computation produces the same value for each of the eigenvalues listed in the table as the bisection with the Sturm sequence computation. It also can be seen that the running time of the scaled Sturm sequence computation is around 1.5 times of that of the Sturm sequence computation.

Bibliography

[1] W. Barth, R. S. Martin, and J. H. Wilkinson, *Calculation of the eigenvalues of a symmetric tridiagonal matrix by the method of bisection*, Numerische Mathematik, 9 (1967), pp 386–393.

[2] K. Chung and W. Yan, *Solving the symmetric tridiagonal eigenvalue problem on hypercubes*, Computers & Mathematics with Applications, 25 (1993), pp 91–96.

[3] J. Demmel, I. Dhillon, and H. Ren, *On the correctness of some bisection-like parallel eigenvalue algorithms in floating point arithmetic*, Electronic Transaction on Numerical Analysis, 3 (1995), pp. 116–149.

[4] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Second ed., The Johns Hopskins University Press, Baltimore, MD, 1989.

[5] R. Gregory and D. Karney, *A Collection of Matrices for Testing Computational Algorithms*, Robert E. Kreger Publishing Company, Huntington, NY, 1978.

[6] T. Y. Li and Z .Zeng, *The laguerre iteration in solving the symmetric tridiagonal eigenproblem, revisited*, SIAM Journal on Scientific Computing, 15 (1994), pp 1145–1173.

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

[8] J. Zhang and D. Friesen, *Parallelizing the computation of one eigenvalue a large symmetric tridiagonal matrix*, Proceedings of the Eighth SIAM Conference on Parallel Processing for Scientific Computing, Minneapolis, March 14-17, 1997.

[9] J. Zhang, *Parallelizing the Computation of Eigenvalues for Real Symmetric Tridiagonal Matrices*, Ph.D. Dissertation, Texas A&M University, College Station, 1999.