SIAM J. MATRIX ANAL. APPL.
Vol. 32, No. 1, pp. 90–114

COMPUTING CHARACTERISTIC POLYNOMIALS FROM EIGENVALUES*

RIZWANA REHMAN† AND ILSE C. F. IPSEN‡

Abstract. This paper concerns the computation of the coefficients c_k of the characteristic polynomial of a real or complex matrix A. We analyze the forward error in the coefficients c_k when they are computed from the eigenvalues of A, as is done by MATLAB's `poly` function. In particular, we derive absolute and relative perturbation bounds for elementary symmetric functions, which we use in turn to derive perturbation bounds for the coefficients c_k with regard to absolute and relative changes in the eigenvalues λ_j of A. We present the so-called Summation Algorithm for computing the coefficients c_k from the eigenvalues λ_j, which is essentially the algorithm used by `poly`. We derive roundoff error bounds and running error bounds for the Summation Algorithm. The roundoff error bounds imply that the Summation Algorithm is forward stable. The running error bounds can be used to estimate the accuracy of the computed coefficients "on the fly," and they tend to be less pessimistic than the roundoff error bounds. Numerical experiments illustrate that our bounds give useful estimates for the accuracy of the coefficients c_k. In particular, the bounds confirm that `poly` computes the coefficients c_k to high relative accuracy if the eigenvalues are positive and given to high relative accuracy.

Key words. elementary symmetric functions, perturbation bounds, roundoff error bounds, determinant

AMS subject classifications. 65F15, 65F40, 65G50, 15A15, 15A18

DOI. 10.1137/100788392

1. Introduction.

The characteristic polynomial of an $n \times n$ real or complex matrix A is defined as

$$\det(zI - A) = z^n + c_1 z^{n-1} + \cdots + c_{n-1} z + c_n,$$

where I is the identity matrix, $c_1 = -\operatorname{trace}(A)$, and $c_n = (-1)^n \det(A)$.

The goal is to analyze the forward error in the coefficients c_k when they are computed from the eigenvalues of A, as is done in the `poly` function of the software package MATLAB.[1] MATLAB's `poly` function first computes the eigenvalues of A with the `eig` function and then determines the coefficients with the so-called Summation Algorithm.

This paper differs from our previous paper [9] because there we derived bounds for the coefficients c_j with regard to changes in the matrix, whereas here we derive bounds with regard to changes in the eigenvalues.

1.1. Main results.

The idea is to relate the coefficients c_k to elementary symmetric functions in the eigenvalues λ_i of A via $c_k = (-1)^k s_k(\lambda)$. The elementary symmetric functions are defined as

$$s_k(\lambda) \equiv \sum_{1 \le i_1 < \cdots < i_k \le n} \lambda_{i_1} \cdots \lambda_{i_k}.$$

*Received by the editors March 11, 2010; accepted for publication (in revised form) November 29, 2010; published electronically February 3, 2011.
http://www.siam.org/journals/simax/32-1/78839.html

†Department of Medicine (111D), VA Medical Center, 508 Fulton Street, Durham, NC 27705 (Rizwana.Rehman@va.gov).

‡Department of Mathematics, North Carolina State University, P.O. Box 8205, Raleigh, NC 27695-8205 (ipsen@ncsu.edu, http://www4.ncsu.edu/∼ipsen/).

[1]MATLAB is a registered trademark of The MathWorks. (http://www.mathworks.com/products/matlab/).

Absolute perturbations. Let $\tilde{\lambda}_i$ be eigenvalues of a matrix whose characteristic polynomial is

$$z^n + \tilde{c}_1 z^{n-1} + \cdots + \tilde{c}_{n-1} z + \tilde{c}_n,$$

and express the absolute perturbation in the eigenvalues as[2] $\epsilon_{abs} \equiv \max_{1 \le i \le n} |\tilde{\lambda}_i - \lambda_i|$. Then the absolute change in the polynomial coefficients with regard to *changes in the eigenvalues* is to first order (Theorem 2.12)

$$|\tilde{c}_k - c_k| \lesssim (n-k+1) s_{k-1}(|\lambda|)\, \epsilon_{abs},$$

where $s_k(|\lambda|)$ is the elementary symmetric function in the absolute values of the eigenvalues. This means if $s_{k-1}(|\lambda|)$ is large, then small absolute perturbations of the eigenvalues can cause a large absolute error in c_k. Hence we can interpret $s_{k-1}(|\lambda|)$ as a first order absolute condition number for c_k with respect to absolute perturbations in the eigenvalues.

To estimate ϵ_{abs} and reveal the sensitivity of the computed eigenvalues, let us consider a diagonalizable matrix $A = Q \Lambda Q^{-1}$ with eigenvalues λ_i and a perturbed matrix $A + E$ with eigenvalues $\tilde{\lambda}_i$. Then the absolute change in the polynomial coefficients with regard to *changes in the matrix* is to first order (Theorem 2.14)

$$|\tilde{c}_k - c_k| \lesssim (n-k+1) s_{k-1}(|\lambda|)\, \|Q\|_2 \|Q^{-1}\|_2\, \|E\|_2.$$

The intermediate computation of the eigenvalues causes the perturbation E to be amplified by the condition number of the eigenvectors Q with respect to inversion. In the special case when A is normal or Hermitian, then $\|Q\|_2 \|Q^{-1}\|_2 = 1$ so that the eigenvalues are insensitive to changes in the matrix, and

$$|\tilde{c}_k - c_k| \lesssim (n-k+1) s_{k-1}(|\lambda|)\, \|E\|_2.$$

Relative perturbations. Let A be nonsingular, and let $\hat{\lambda}_i$ be eigenvalues of a matrix whose characteristic polynomial is

$$z^n + \hat{c}_1 z^{n-1} + \cdots + \hat{c}_{n-1} z + \hat{c}_n.$$

Express the relative eigenvalue perturbation as $\epsilon_{rel} \equiv \max_{1 \le i \le n} |\hat{\lambda}_i - \lambda_i| / |\lambda_i|$. Then the relative change in the polynomial coefficients with regard to *changes in the eigenvalues* is to first order (Theorem 2.13)

$$\frac{|\hat{c}_k - c_k|}{|c_k|} \lesssim k \frac{s_k(|\lambda|)}{|c_k|}\, \epsilon_{rel}.$$

This means if $k s_k(|\lambda|)/|c_k|$ is large, then small relative perturbations of the eigenvalues can cause a large relative error in c_k. Hence we can interpret $k s_k(|\lambda|)/|c_k|$ as a first order relative condition number for c_k with respect to relative perturbations in the eigenvalues. In particular, if all $\lambda_i > 0$, then $s_k(|\lambda|) = |c_k|$ and is to first order

$$\frac{|\hat{c}_k - c_k|}{|c_k|} \lesssim k\, \epsilon_{rel}.$$

[2]To simplify the exposition, we assume in this section that the eigenvalues are real and labeled in increasing or decreasing order.

This means if all eigenvalues are positive, then there is no cancellation in the computation of the coefficients c_k, and they are well-conditioned in the relative sense with regard to relative changes in the eigenvalues.

To see the effect of the computed eigenvalues, we choose a specific value for ϵ_{rel} for a normal matrix A with eigenvalues λ_i and a perturbed matrix $A+E$ with eigenvalues $\hat{\lambda}_i$. Then the relative change in the polynomial coefficients with regard to *changes in the matrix* is to first order (Theorem 2.15)

$$\frac{|\hat{c}_k - c_k|}{|c_k|} \leq kn^2 \frac{s_k(|\lambda|)}{|c_k|} \, \|A^{-1}\|_2 \|E\|_2.$$

Hence the eigenvalue computation can amplify the perturbation in the matrix by a factor of $n^2\|A^{-1}\|_2$.

Roundoff error bounds. We show that the Summation Algorithm, which computes the polynomial coefficients from the eigenvalues and is essentially the algorithm used by MATLAB's `poly` function, is forward stable (Remark 4.1). As a result, the roundoff error bounds for the coefficients are similar to the perturbation bounds.

Assume the eigenvalues are computed to absolute accuracy ϵ_{abs}, and denote by $\mathrm{fl}[\tilde{c}_k]$ the coefficient $\tilde{c}_k$ computed in floating point arithmetic. Then to first order (Theorem 4.9),

$$|\,\mathrm{fl}[\tilde{c}_k] - c_k| \lesssim (n-k+1)s_{k-1}(|\lambda|)\,\epsilon_{abs} + s_k(|\lambda|)\,\gamma_{2n},$$

where u is the unit roundoff, and $\gamma_{2n} \equiv 2nu/(1-2nu)$. This means in floating point arithmetic, there are two amplification factors: $s_{k-1}(|\lambda|)$ and $s_k(|\lambda|)$.

If the eigenvalues are computed to relative accuracy ϵ_{rel} and $\mathrm{fl}[\hat{c}_k]$ denotes the coefficient $\hat{c}_k$ computed in floating point arithmetic, then to first order (Theorem 4.10),

$$\frac{|\,\mathrm{fl}[\hat{c}_k] - c_k|}{|c_k|} \lesssim \frac{s_k(|\lambda|)}{|c_k|}\,\left(\gamma_{2n} + k\epsilon_{rel}\right).$$

Hence in floating point arithmetic, relative perturbations in the eigenvalues are amplified by $s_k(|\lambda|)/|c_k|$. In particular, if all eigenvalues are positive, then

$$\frac{|\,\mathrm{fl}[\hat{c}_k] - c_k|}{|c_k|} \lesssim \gamma_{2n} + k\epsilon_{rel}. \tag{1.1}$$

As before, one can replace ϵ_{abs} and ϵ_{rel} in the roundoff error bounds by perturbation bounds for particular classes of matrices. We also present running error bounds (Theorem 4.8) to estimate the accuracy of the computed coefficients "on the fly." These bounds tend to be less pessimistic than the worst case roundoff error bounds, because they use intermediate quantities from the Summation Algorithm and can account for some degree of cancellation due to subtractions.

Numerical experiments. The experiments (section 5) illustrate that our bounds give useful estimates for the accuracy of the coefficients c_k. In particular, the experiments confirm the bound (1.1) which implies that `poly` should compute the coefficients c_k to high relative accuracy if the eigenvalues are positive and given to high relative accuracy, because in this case `poly` encounters no cancellation due to subtractions.

We illustrate that `poly` can compute the coefficients c_k to high accuracy, regardless of whether the eigenvalues are well separated or are clustered (section 5.2). This

happens, for instance, with Hermitian positive definite matrices because their eigenvalues λ_i are well-conditioned and positive so that the coefficients c_k are well-conditioned with regard to changes in the λ_i. In contrast, though, `poly` does not necessarily compute accurate coefficients just because the eigenvalues are accurate (sections 5.3 and 5.4). This happens, for instance, with Hermitian indefinite matrices whose eigenvalues λ_i are well-conditioned, but the coefficients c_k are ill-conditioned due to cancellation from subtractions. Of course, when the eigenvalues are ill-conditioned, then `poly` cannot be expected to compute accurate coefficients c_k from these ill-conditioned eigenvalues (sections 5.1 and 5.6).

1.2. Overview. In section 2 we derive absolute and relative perturbation bounds for elementary symmetric functions, and for the coefficients c_k with regard to absolute and relative changes in the eigenvalues. In section 3 we present the Summation Algorithm for computing the coefficients c_k from the eigenvalues λ_j, and in section 4 we derive roundoff error bounds and running error bounds for the Summation Algorithm. Numerical experiments in section 5 illustrate the performance of the bounds.

1.3. Notation. We denote by $\lambda = (\lambda_1 \quad \dots \quad \lambda_n)$ a vector of real or complex numbers λ_i. The absolute value applies componentwise, i.e., $|\lambda| \equiv (|\lambda_1| \quad \dots \quad |\lambda_n|)$. The inequality $\lambda > 0$ means that $\lambda_i > 0$ for all $1 \le i \le n$. The identity matrix is denoted by I, and the superscript $*$ denotes the conjugate transpose. The expression $\mathrm{diag}(\lambda_1, \dots, \lambda_n)$ denotes an $n \times n$ diagonal matrix with diagonal elements λ_j.

2. Perturbation bounds. We start by deriving perturbation bounds for elementary symmetric functions. Absolute perturbation bounds are derived in section 2.1, and relative bounds are derived in section 2.2. These bounds show that the sensitivity of the elementary symmetric functions to changes in the inputs λ can be expressed in terms of elementary symmetric functions of $|\lambda|$, i.e., the *absolute value* of the inputs. The bounds are then used in section 2.3 to produce perturbation bounds for characteristic polynomials with regard to changes in the eigenvalues. In section 2.4 we customize these bounds to particular classes of matrices to reveal the effect of computed eigenvalues.

DEFINITION 2.1. *For a vector of n complex or real numbers $\lambda = (\lambda_1 \quad \dots \quad \lambda_n)$, the kth elementary symmetric function is defined as*

$$s_0(\lambda) \equiv 1, \qquad s_k(\lambda) \equiv \sum_{1 \le i_1 < \cdots < i_k \le n} \lambda_{i_1} \cdots \lambda_{i_k}, \quad 1 \le k \le n.$$

If the inputs are eigenvalues of a matrix, then the elementary symmetric functions are, up to a sign, equal to the coefficients of the characteristic polynomial.

LEMMA 2.2 (Theorem 1.2.12 in [8]). *If A is a real or complex $n \times n$ matrix with eigenvalues $\lambda_1, \dots, \lambda_n$, then for $1 \le k \le n$*

1. *$s_k(\lambda) = (-1)^k c_k$;*
2. *$s_k(\lambda)$ is the sum of all $k \times k$ principal minors of A.*

2.1. Absolute perturbations. We consider absolute perturbations of the inputs,

$$\tilde{\lambda} = (\tilde{\lambda}_1 \quad \dots \quad \tilde{\lambda}_n), \quad \text{where} \quad \tilde{\lambda}_i \equiv \lambda_i + \epsilon_i, \quad 1 \le i \le n. \tag{2.1}$$

We use the following approach for bounding the absolute error in the elementary symmetric function $s_k(\tilde{\lambda})$. From the inputs λ_i and $\tilde{\lambda}_i$, we construct diagonal matrices, and then we apply the perturbation bounds for characteristic polynomials with regard

to absolute changes in the matrix from our previous paper [9]. Combined with the relation between elementary symmetric functions and characteristic polynomials in Lemma 2.2, this yields new perturbation bounds for elementary symmetric functions with regard to absolute changes in the inputs. These new bounds are expressed in terms of elementary functions of $|\lambda|$, the magnitude of the inputs.

THEOREM 2.3 (Theorem 3.5 in [9]). *Let A and $A+E$ be $n \times n$ complex matrices with respective characteristic polynomials*

$$\begin{aligned} \det(zI - A) &= z^n + c_1 z^{n-1} + \cdots + c_{n-1} z + c_n, \\ \det(zI - (A+E)) &= z^n + \tilde{c}_1 z^{n-1} + \cdots + \tilde{c}_{n-1} z + \tilde{c}_n. \end{aligned}$$

If A is normal (or Hermitian), then

$$|\tilde{c}_k - c_k| \leq \sum_{i=1}^{k} \binom{n-k+i}{i} s_{k-i}\left(|\lambda|\right) \|E\|_2^i, \qquad 1 \leq k \leq n.$$

With the help of Theorem 2.3 we bound the absolute error $|s_k(\tilde{\lambda}) - s_k(\lambda)|$ in terms of elementary symmetric functions applied to the absolute values, $|\lambda|$.

THEOREM 2.4 (absolute perturbations). *If $\tilde{\lambda}$ is an absolute perturbation* (2.1)*, then*

$$|s_k(\tilde{\lambda}) - s_k(\lambda)| \leq \sum_{i=1}^{k} \binom{n-k+i}{i} s_{k-i}(|\lambda|)\, \epsilon_{abs}^i, \qquad 1 \leq k \leq n,$$

where $\epsilon_{abs} \equiv \max_{1 \leq i \leq n} |\epsilon_i|$.

Proof. The first part of Lemma 2.2 implies $|s_k(\tilde{\lambda}) - s_k(\lambda)| = |\tilde{c}_k - c_k|$. Applying Theorem 2.3 to the diagonal matrices $A \equiv \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ and $E \equiv \operatorname{diag}(\epsilon_1, \ldots, \epsilon_n)$ and realizing that $\|E\|_2 = \epsilon_{abs}$ gives the desired bound. ∎

Theorem 2.4 bounds the absolute error in the kth elementary symmetric function $s_k(\tilde{\lambda})$ in terms of the "preceding" elementary symmetric functions $s_1, \ldots, s_{k-1}$, but applied to $|\lambda|$. In particular, the first elementary symmetric function $s_1(\lambda) = \lambda_1 + \cdots + \lambda_n$ is well-conditioned in the absolute sense for sufficiently small n because

$$|s_1(\tilde{\lambda}) - s_1(\lambda)| \leq n \epsilon_{abs}.$$

For the remaining symmetric functions and $\epsilon_{abs} < 1$, Theorem 2.4 implies the first order absolute bounds

$$|s_k(\tilde{\lambda}) - s_k(\lambda)| \leq (n-k+1)\, s_{k-1}(|\lambda|)\, \epsilon_{abs} + \mathcal{O}(\epsilon_{abs}^2), \qquad 2 \leq k \leq n-1. \tag{2.2}$$

This suggests that small absolute perturbations in λ can cause large absolute errors in $s_k(\tilde{\lambda})$ if $s_{k-1}(|\lambda|)$ is large. For the last elementary symmetric function $s_n(\lambda) = \lambda_1 \cdots \lambda_n$, we obtain

$$|s_n(\tilde{\lambda}) - s_n(\lambda)| \leq s_{n-1}(|\lambda|)\, \epsilon_{abs} + \mathcal{O}(\epsilon_{abs}^2).$$

A similar result is derived in [5, Lemma 3] by means of an inequality due to Mitrinović [11, page 315].

2.2. Relative perturbations. We consider relative perturbations of the inputs

$$\hat{\lambda} = (\hat{\lambda}_1 \quad \dots \quad \hat{\lambda}_n), \quad \text{where} \quad \hat{\lambda}_i \equiv \lambda_i(1+\epsilon_i), \quad 1 \le i \le n. \tag{2.3}$$

The approach for bounding the relative error in $s(\hat{\lambda})$ is the same as in section 2.1. From the inputs λ_i and $\hat{\lambda}_i$, we construct diagonal matrices, and then we apply the perturbation bounds for characteristic polynomials with regard to relative changes in the *matrix* from our previous paper [9]. Combined with the relation between elementary symmetric functions and characteristic polynomials in Lemma 2.2, this yields new perturbation bounds for elementary symmetric functions with regard to relative changes in the inputs. As in section 2.1, these new bounds are expressed in terms of elementary functions applied to $|\lambda|$, the magnitude of the inputs.

We start with a bound for the last elementary function, and then we use it to derive perturbation bounds for the remaining functions.

2.2.1. The last elementary symmetric function. We use the relative expansion for the determinant below to derive an expression for $s_n(\hat{\lambda}) - s_n(\lambda)$.

THEOREM 2.5 (Theorem 2.13 in [9]). *Let A and E be $n \times n$ complex matrices. If A is nonsingular, then*

$$\frac{\det(A+E) - \det(A)}{\det(A)} = \det(A^{-1}E) + S_1 + \cdots + S_{n-1},$$

where

$$S_k \equiv \sum_{1 \le i_1 < \cdots < i_k \le n} \det((A^{-1}E)_{i_1 \dots i_k}), \qquad 1 \le k \le n-1.$$

Here $((A^{-1}E)_{i_1 \dots i_k})$ is the principal submatrix of order $n-k$ obtained by removing rows and columns $i_1, \dots, i_k$ from $A^{-1}E$.

With the help of Theorem 2.5 we derive an expression for the error $s_n(\hat{\lambda}) - s_n(\lambda)$ in terms of elementary symmetric functions of the relative errors $\epsilon = (\epsilon_1 \quad \dots \quad \epsilon_n)$.

THEOREM 2.6 (error expansion for $s_n(\hat{\lambda})$). *If $\hat{\lambda}$ is a relative perturbation* (2.3)*, then*

$$s_n(\hat{\lambda}) - s_n(\lambda) = s_n(\lambda) \sum_{i=1}^{n} s_i(\epsilon).$$

Proof. If $\lambda_i = 0$ for some i, then $s_n(\lambda) = \lambda_1 \cdots \lambda_n = 0$. Moreover, $\hat{\lambda}_i = \lambda(1+\epsilon_i) = 0$ so that $s_n(\hat{\lambda}) = 0$, and the desired result holds.

Now assume that $\lambda_i \neq 0$ for all $1 \le i \le n$. Define the diagonal matrices $A \equiv \operatorname{diag}(\lambda_1, \dots, \lambda_n)$ and $E \equiv \operatorname{diag}(\lambda_1\epsilon_1, \dots, \lambda_n\epsilon_n)$ so that $\det(A) = s_n(\lambda)$, and $A^{-1}E = \operatorname{diag}(\epsilon_1, \dots, \epsilon_n)$ with $\det(A^{-1}E) = s_n(\epsilon)$. Applying Theorem 2.5 to A and $A+E$ gives

$$s_n(\hat{\lambda}) - s_n(\lambda) = s_n(\lambda)\left(s_n(\epsilon) + S_1 + \cdots + S_{n-1}\right).$$

The second part of Lemma 2.2 implies $S_k = s_{n-k}(\epsilon)$, $1 \le k \le n-1$. Hence $\sum_{k=1}^{n-1} S_k = \sum_{k=1}^{n-1} s_k(\epsilon)$. ∎

Theorem 2.6 implies the following relative bounds for $s_n(\hat{\lambda})$.

COROLLARY 2.7 (relative perturbation bounds for $s_n(\hat{\lambda})$). *If $\hat{\lambda}$ is a relative perturbation* (2.3)*, then*

$$|s_n(\hat{\lambda}) - s_n(\lambda)| \le |s_n(\lambda)| \sum_{i=1}^{n} \binom{n}{i} \epsilon_{rel}^i = |s_n(\lambda)| \left[(1+\epsilon_{rel})^n - 1\right],$$

where $\epsilon_{rel} \equiv \max_{1\leq i\leq n} |\epsilon_i|$. *If also* $n\epsilon_{rel} < 1$, *then*

$$|s_n(\hat{\lambda}) - s_n(\lambda)| \leq \frac{n\epsilon_{rel}}{1 - n\epsilon_{rel}} |s_n(\lambda)|.$$

Proof. In the right-hand-side expression of Theorem 2.6, each $s_i(\epsilon)$ is a sum of $\binom{n}{i}$ terms, where each term is a product of i factors ϵ_j, $1 \leq i, j \leq n$. Therefore, $|s_i(\epsilon)| \leq \binom{n}{i}\epsilon_{rel}^i$. This gives the first inequality:

$$|s_n(\hat{\lambda}) - s_n(\lambda)| \leq |s_n(\lambda)| \sum_{i=1}^{n} \binom{n}{i} \epsilon_{rel}^i = |s_n(\lambda)| \left[(1+\epsilon_{rel})^n - 1\right].$$

A further upper bound produces the second inequality as follows. If $n\epsilon_{rel} < 1$, we can replace the finite sum by an infinite geometric series,

$$(1+\epsilon_{rel})^n = \sum_{j=0}^{n} \binom{n}{j} \epsilon_{rel}^j \leq \sum_{j=0}^{n} n^j \epsilon_{rel}^j \leq \sum_{j=0}^{\infty} (n\epsilon_{rel})^j = \frac{1}{1 - n\epsilon_{rel}}.$$

Hence

$$|s_n(\lambda)| \left[(1+\epsilon_{rel})^n - 1\right] \leq |s_n(\lambda)| \left[\frac{1}{1 - n\epsilon_{rel}} - 1\right] = |s_n(\lambda)| \frac{n\epsilon_{rel}}{1 - n\epsilon_{rel}}. \qquad \square$$

For $\epsilon_{rel} < 1$, Corollary 2.7 implies the first order relative bound

$$|s_n(\hat{\lambda}) - s_n(\lambda)| \leq |s_n(\lambda)| \, n\epsilon_{rel} + \mathcal{O}(\epsilon_{rel}^2).$$

This means if n is sufficiently small, then $s_n(\lambda)$ is well-conditioned with respect to relative perturbations in λ.

2.2.2. All elementary symmetric functions. With the help of Theorem 2.6 we derive an expansion for the error $s_k(\hat{\lambda}) - s_k(\lambda)$ in terms of elementary functions of subsets of relative errors ϵ_i. The example below illustrates the expansion.

Example 2.1 ($n = 4$). Expansion of $s_3(\hat{\lambda}) - s_3(\lambda)$ for $\lambda = (\lambda_1 \quad \dots \quad \lambda_4)$).

Define the diagonal matrix $A \equiv \operatorname{diag}(\lambda_1, \dots, \lambda_4)$. The second part of Lemma 2.2 implies that $s_3(\lambda)$ is a sum of 3×3 principal minors of A. That is,

$$s_3(\lambda) = \lambda_1\lambda_2\lambda_3 + \lambda_1\lambda_2\lambda_4 + \lambda_1\lambda_3\lambda_4 + \lambda_2\lambda_3\lambda_4.$$

Applying Theorem 2.6 to each of these four products gives

$$\begin{aligned} s_3(\hat{\lambda}) - s_3(\lambda) = {} & \lambda_1\lambda_2\lambda_3 \left[(\epsilon_1 + \epsilon_2 + \epsilon_3) + (\epsilon_1\epsilon_2 + \epsilon_1\epsilon_3 + \epsilon_2\epsilon_3) + \epsilon_1\epsilon_2\epsilon_3\right] \\ & + \lambda_1\lambda_2\lambda_4 \left[(\epsilon_1 + \epsilon_2 + \epsilon_4) + (\epsilon_1\epsilon_2 + \epsilon_1\epsilon_4 + \epsilon_2\epsilon_4) + \epsilon_1\epsilon_2\epsilon_4\right] \\ & + \lambda_1\lambda_3\lambda_4 \left[(\epsilon_1 + \epsilon_3 + \epsilon_4) + (\epsilon_1\epsilon_3 + \epsilon_1\epsilon_4 + \epsilon_3\epsilon_4) + \epsilon_1\epsilon_3\epsilon_4\right] \\ & + \lambda_2\lambda_3\lambda_4 \left[(\epsilon_2 + \epsilon_3 + \epsilon_4) + (\epsilon_2\epsilon_3 + \epsilon_2\epsilon_4 + \epsilon_3\epsilon_4) + \epsilon_2\epsilon_3\epsilon_4\right]. \end{aligned}$$

The ϵ terms on the right-hand side are elementary symmetric functions of three elements of ϵ. For instance, the right-hand side of

$$\hat{\lambda}_1\hat{\lambda}_2\hat{\lambda}_4 - \lambda_1\lambda_2\lambda_4 = \lambda_1\lambda_2\lambda_4 \left[(\epsilon_1 + \epsilon_2 + \epsilon_4) + (\epsilon_1\epsilon_2 + \epsilon_1\epsilon_4 + \epsilon_2\epsilon_4) + \epsilon_1\epsilon_2\epsilon_4\right]$$

contains elementary symmetric functions of ϵ_1, ϵ_2, and ϵ_4. Set

$$s_1^{(124)}(\epsilon) \equiv \epsilon_1 + \epsilon_2 + \epsilon_4, \quad s_2^{(124)}(\epsilon) \equiv \epsilon_1\epsilon_2 + \epsilon_1\epsilon_4 + \epsilon_2\epsilon_4, \quad s_3^{(124)}(\epsilon) \equiv \epsilon_1\epsilon_2\epsilon_4.$$

Then we can write

$$\hat{\lambda}_1\hat{\lambda}_2\hat{\lambda}_4 - \lambda_1\lambda_2\lambda_4 = \lambda_1\lambda_2\lambda_4 \left[s_1^{(124)}(\epsilon) + s_2^{(124)}(\epsilon) + s_3^{(124)}(\epsilon)\right].$$

Repeating this for all four products in $s_3(\hat{\lambda}) - s_3(\lambda)$ gives

$$\begin{aligned} s_3(\hat{\lambda}) - s_3(\lambda) = {} & \lambda_1\lambda_2\lambda_3 \left[s_1^{(123)}(\epsilon) + s_2^{(123)}(\epsilon) + s_3^{(123)}(\epsilon)\right] \\ & + \lambda_1\lambda_2\lambda_4 \left[s_1^{(124)}(\epsilon) + s_2^{(124)}(\epsilon) + s_3^{(124)}(\epsilon)\right] \\ & + \lambda_1\lambda_3\lambda_4 \left[s_1^{(134)}(\epsilon) + s_2^{(134)}(\epsilon) + s_3^{(134)}(\epsilon)\right] \\ & + \lambda_2\lambda_3\lambda_4 \left[s_1^{(234)}(\epsilon) + s_2^{(234)}(\epsilon) + s_3^{(234)}(\epsilon)\right]. \end{aligned}$$

In order to extend this example to any n, we introduce notation for general elementary symmetric functions of subsets of elements.

DEFINITION 2.8. *For a vector of n complex numbers $\lambda = (\lambda_1 \quad \dots \quad \lambda_n)$ and indices $1 \le i_1 < \cdots < i_k \le n$, the jth elementary function of the k-element subvector $(\lambda_{i_1} \quad \dots \quad \lambda_{i_k})$ is defined as*

$$s_j^{(i_1 \cdots i_k)}(\lambda) \equiv s_j(\lambda_{i_1} \quad \dots \quad \lambda_{i_k}), \qquad 1 \le j \le k.$$

In particular, $s_j^{(1\dots n)}(\lambda) \equiv s_j(\lambda)$, $1 \le j \le n$.

Now we are ready to extend Theorem 2.6 to the remaining elementary symmetric functions.

THEOREM 2.9 (error expansion for all $s_k(\hat{\lambda})$). *If $\hat{\lambda}$ is a relative perturbation* (2.3)*, then*

$$s_k(\hat{\lambda}) - s_k(\lambda) = \sum_{1 \le i_1 < \cdots < i_k \le n} \lambda_{i_1} \cdots \lambda_{i_k} \sum_{j=1}^{k} s_j^{(i_1 \dots i_k)}(\epsilon), \qquad 1 \le k \le n.$$

Proof. Define the diagonal matrix $A \equiv \operatorname{diag}(\lambda_1, \dots, \lambda_n)$. According to the second part of Lemma 2.2, each $s_k(\lambda)$ is a sum of $\binom{n}{k}$ principal minors of order k. Since A is diagonal, such a principal minor is a product of k numbers, $\lambda_{i_1} \cdots \lambda_{i_k}$. Applying Theorem 2.6 to the product $\hat{\lambda}_{i_1} \cdots \hat{\lambda}_{i_k}$ gives

$$\hat{\lambda}_{i_1} \cdots \hat{\lambda}_{i_k} - \lambda_{i_1} \cdots \lambda_{i_k} = \lambda_{i_1} \cdots \lambda_{i_k} \sum_{j=1}^{k} s_j^{(i_1 \dots i_k)}(\epsilon).$$

Summing up these expansions for all principal minors gives the desired result. ∎

The connection to Theorem 2.6 may be even clearer if we view the products as elementary symmetric functions, i.e., $\lambda_{i_1} \cdots \lambda_{i_k} = s_k^{(i_1 \dots i_k)}(\lambda)$, and express Theorem 2.9 as

$$s_k(\hat{\lambda}) - s_k(\lambda) = \sum_{1 \le i_1 < \cdots < i_k \le n} s_k^{(i_1 \dots i_k)}(\lambda) \sum_{j=1}^{k} s_j^{(i_1 \dots i_k)}(\epsilon), \qquad 1 \le k \le n.$$

Theorem 2.9 implies the following bounds for the relative error in $s_k(\hat{\lambda})$.

COROLLARY 2.10 (relative perturbation bounds for all $s_k(\hat{\lambda})$). *If $\hat{\lambda}$ is a relative perturbation* (2.3)*, then for $1 \leq k \leq n$*

$$|s_k(\hat{\lambda}) - s_k(\lambda)| \leq s_k(|\lambda|) \sum_{j=1}^{k} \binom{k}{j} \epsilon_{rel}^j = s_k(|\lambda|) \left[(1+\epsilon_{rel})^k - 1\right],$$

where $\epsilon_{rel} \equiv \max_{1\leq i\leq n} |\epsilon_i|$. If also $n\epsilon_{rel} < 1$, then

$$|s_k(\hat{\lambda}) - s_k(\lambda)| \leq \frac{k\epsilon_{rel}}{1-k\epsilon_{rel}}\, s_k(|\lambda|), \qquad 1 \leq k \leq n.$$

Proof. Applying the triangle inequality to the expression in Theorem 2.9 gives

$$|s_k(\hat{\lambda}) - s_k(\lambda)| \leq \sum_{1\leq i_1<\cdots<i_k\leq n} |\lambda_{i_1}|\cdots|\lambda_{i_k}| \sum_{j=1}^{k} |s_j^{(i_1\ldots i_k)}(\epsilon)|, \qquad 1 \leq k \leq n.$$

Bounding the elementary symmetric functions by $|s_j^{(i_1\ldots i_k)}(\epsilon)| \leq \binom{k}{j}\epsilon_{rel}^j$ and summing up all the bounds yields

$$\begin{aligned}\sum_{1\leq i_1<\cdots<i_k\leq n} |\lambda_{i_1}|\cdots|\lambda_{i_k}| \sum_{j=1}^{k} |s_j^{(i_1\ldots i_k)}(\epsilon)| &\leq \sum_{1\leq i_1<\cdots<i_k\leq n} |\lambda_{i_1}|\cdots|\lambda_{i_k}| \sum_{j=1}^{k} \binom{k}{j}\epsilon_{rel}^j \\ &= s_k(|\lambda|) \sum_{j=1}^{k} \binom{k}{j}\epsilon_{rel}^j. \qquad \square\end{aligned}$$

If $s_k(\lambda) \neq 0$, then Corollary 2.10 implies the relative error bound

$$\frac{|s_k(\hat{\lambda}) - s_k(\lambda)|}{|s_k(\lambda)|} \leq \frac{s_k(|\lambda|)}{|s_k(\lambda)|} \left[(1+\epsilon_{rel})^k - 1\right], \qquad 1 \leq k \leq n.$$

This suggests that small relative perturbations in λ can cause large relative errors in $s_k(\hat{\lambda})$ if $s_k(|\lambda|) \gg |s_k(\lambda)|$, i.e., if there are many sign changes in the inputs. The only situation when we can expect a good bound is if all inputs have the same sign so that no cancellation occurs. This agrees with [2], where it was shown that $s_k(\lambda)$ is well-conditioned in the relative sense if all elements of λ are positive.

COROLLARY 2.11 (Proposition 7.1 in [2]). *If, in addition to the assumptions of Corollary* 2.10*, also $\lambda > 0$, then*

$$\frac{|s_k(\hat{\lambda}) - s_k(\lambda)|}{s_k(\lambda)} \leq \frac{k\epsilon_{rel}}{1-k\epsilon_{rel}}, \qquad 1 \leq k \leq n.$$

For positive inputs Corollary 2.11 implies the following first order bound:

$$\frac{|s_k(\hat{\lambda}) - s_k(\lambda)|}{s_k(\lambda)} \leq k\epsilon_{rel} + \mathcal{O}\left(\epsilon_{rel}^2\right), \qquad 1 \leq k \leq n.$$

2.3. Perturbation bounds for characteristic polynomials. We use the bounds for elementary functions in sections 2.1 and 2.2 to derive perturbation bounds for characteristic polynomials with regard to changes in the eigenvalues. Note that this is different from our previous paper [9] where the perturbation bounds were derived with regard to changes in the matrix.

Let A be an $n \times n$ complex matrix with the characteristic polynomial

$$\det(zI - A) = z^n + c_1 z^{n-1} + \cdots + c_{n-1} z + c_n$$

and eigenvalues $\lambda_1, \ldots, \lambda_n$.

First we bound the absolute error in the coefficients in terms of the absolute error in the eigenvalues.

THEOREM 2.12 (absolute perturbations). *Let $\tilde{\lambda}_i \equiv \lambda_i + \epsilon_i$ be eigenvalues of a matrix whose characteristic polynomial is $z^n + \tilde{c}_1 z^{n-1} + \cdots + \tilde{c}_{n-1} z + \tilde{c}_n$.*

If $\epsilon_{abs} \equiv \max_{1 \le i \le n} |\epsilon_i| < 1$, then

$$|\tilde{c}_k - c_k| \le (n - k + 1) s_{k-1}(|\lambda|) \epsilon_{abs} + \mathcal{O}\big(\epsilon_{abs}^2\big), \qquad 1 \le k \le n.$$

Proof. Lemma 2.2 implies $|\tilde{c}_k - c_k| = |s_k(\tilde{\lambda}) - s_k(\lambda)|$. The bounds follow from Theorem 2.4. ∎

Hence we can interpret $s_{k-1}(|\lambda|)$ as a first order absolute condition number for c_k with respect to absolute perturbations in the eigenvalues of A. Next we bound the relative error in the coefficients in terms of the relative error in the eigenvalues.

THEOREM 2.13 (relative perturbations). *Let $\hat{\lambda}_i \equiv \lambda_i(1 + \epsilon_i)$ be eigenvalues of a matrix whose characteristic polynomial is $z^n + \hat{c}_1 z^{n-1} + \cdots + \hat{c}_{n-1} z + \hat{c}_n$, and let $\epsilon_{rel} \equiv \max_{1 \le i \le n} |\epsilon_i|$.*

If $n\epsilon_{rel} < 1$ and $c_k \neq 0$, then

$$\frac{|\hat{c}_k - c_k|}{|c_k|} \le \frac{k\epsilon_{rel}}{1 - k\epsilon_{rel}} \frac{s_k(|\lambda|)}{|c_k|}, \qquad 1 \le k \le n.$$

If, in addition, $\lambda_i > 0$ for $1 \le i \le n$, then

$$\frac{|\hat{c}_k - c_k|}{|c_k|} \le \frac{k\epsilon_{rel}}{1 - k\epsilon_{rel}}, \qquad 1 \le k \le n.$$

Proof. The first bound follows from Lemma 2.2 and Corollary 2.10, and the second one follows from Corollary 2.11. ∎

Hence we can interpret $k s_k(|\lambda|)/|c_k|$ as a first order relative condition number for c_k with respect to relative perturbations in the eigenvalues of A.

2.4. Revealing eigenvalue sensitivity. We present two bounds that illustrate the effect of computed eigenvalues. These bounds are more specialized than those in section 2.3, because they apply to particular classes of matrices, and break down the eigenvalue errors ϵ_{abs} and ϵ_{rel} into concrete expressions that reflect the sensitivity of the eigenvalues to perturbations in the matrix.

The first bound is an absolute perturbation bound for diagonalizable matrices. It depends on the condition number of the eigenvectors with respect to inversion and reflects the sensitivity of the eigenvalues to perturbations in the matrix.

THEOREM 2.14 (diagonalizable matrices). *Let $A = Q\Lambda Q^{-1}$ be diagonalizable, where Λ is diagonal with diagonal elements λ_i, and let $A + E$ have eigenvalues $\tilde{\lambda}_i$ and a characteristic polynomial with coefficients $\tilde{c}_k$.*

If $\rho_{abs} \equiv n \, \|Q^{-1}\|_2 \|Q\|_2 \, \|E\|_2 < 1$, then

$$|\tilde{c}_k - c_k| \le (n - k + 1) s_{k-1}(|\lambda|) \, \rho_{abs} + \mathcal{O}(\rho_{abs}^2), \qquad 1 \le k \le n.$$

Proof. From [13, Theorem 2.4] follows that there exists a permutation τ of $\{1, \ldots, n\}$ so that $\epsilon_{abs} \equiv \max_{1 \le i \le n} |\tilde{\lambda}_{\tau(i)} - \lambda_i| \le \rho_{abs}$. Use this bound in Theorem 2.12. ∎

Below is a relative error bound for normal matrices.

THEOREM 2.15 (normal matrices). *Let A be nonsingular and normal with eigenvalues λ_i, and let $A + E$ have eigenvalues $\hat{\lambda}_i$ and a characteristic polynomial with coefficients $\hat{c}_k$.*

If $n\rho_{rel} < 1$, where $\rho_{rel} \equiv n\|A^{-1}\|_2\|E\|_2$, and $c_k \neq 0$, then

$$\frac{|\hat{c}_k - c_k|}{|c_k|} \le \frac{k\rho_{rel}}{1 - k\rho_{rel}} \frac{s_k(|\lambda|)}{|c_k|}, \qquad 1 \le k \le n.$$

Proof. From [10, Corollary 3.3] follows that there exists a permutation τ of $\{1, \ldots, n\}$ so that

$$\epsilon_{rel} \equiv \max_{1 \le i \le n} \frac{|\hat{\lambda}_{\tau(i)} - \lambda_i|}{|\lambda_i|} \le \rho.$$

Use this bound in Theorem 2.13. ∎

More bounds of this type can be found in [12, section 5.3] for matrices that are not diagonalizable, normal, Hermitian, and totally nonnegative or are given in terms of a symmetric rank revealing decomposition.

3. The Summation Algorithm. Now that we have performed the perturbation analysis, we can go about computing the elementary symmetric functions $s_k(\lambda)$. Given n input numbers $\lambda_1, \ldots, \lambda_n$, the kth function $s_k(\lambda)$ consists of $\binom{n}{k}$ summands. Therefore, straightforward computation of $s_k(\lambda)$ is very expensive. We need a more efficient, yet numerically stable, algorithm.

Baker and Harwell [1] present a collection of algorithms that includes the Difference Algorithm, the Summation Algorithm, and the Grouping Property Algorithm. We focus on the Summation Algorithm because essentially the same algorithm is used by MATLAB's `poly` function to compute the coefficients of a characteristic polynomial from the eigenvalues of the corresponding matrix; see section 5. To describe the Summation Algorithm, we introduce the following abbreviation.

DEFINITION 3.1. *For a vector of n complex or real numbers $\lambda = (\lambda_1 \quad \ldots \quad \lambda_n)$ and an index $1 \le i \le n$, the kth elementary function of the leading i-element subvector $(\lambda_1 \quad \ldots \quad \lambda_i)$ is defined as*

$$s_k^{(i)} \equiv s_k(\lambda_1 \quad \ldots \quad \lambda_i), \qquad 1 \le k \le i.$$

In particular, $s_k^{(n)} \equiv s_k(\lambda)$, $1 \le k \le n$.

The Summation Algorithm, represented by Algorithm 1 below, computes the elementary symmetric functions recursively as in [3] and [4, page 250].

Algorithm 1 requires $\frac{n(n-1)}{2}$ multiplications and $\frac{n(n-1)}{2} + (n+1)$ additions [1]. We illustrate how Algorithm 1 works by applying it to determine the elementary symmetric functions for $\lambda = (\lambda_1 \quad \ldots \quad \lambda_4)$

Algorithm 1 SUMMATION ALGORITHM

Input: $\lambda = \begin{pmatrix}\lambda_1 & \dots & \lambda_n\end{pmatrix}$
Output: Elementary symmetric functions $s_k(\lambda)$

$s_0^{(l)} := 1$, $1 \le l \le n-1$, and $s_k^{(l)} := 0$, $k > l$

$s_1^{(1)} := \lambda_1$
for $i = 2$ to n **do**
  for $k = 1$ to i **do**
    $s_k^{(i)} := s_k^{(i-1)} + \lambda_i\, s_{k-1}^{(i-1)}$
  end for
end for
{Now $s_k(\lambda) = s_k^{(n)}$}

$$
\begin{array}{lll}
i=1: & & s_1^{(1)} = \lambda_1 \\
i=2: & & s_1^{(2)} = s_1^{(1)} + \lambda_2\, s_0^{(1)} = \lambda_1 + \lambda_2 \\
 & & s_2^{(2)} = s_2^{(1)} + \lambda_2\, s_1^{(1)} = \lambda_2\lambda_1 \\
i=3: & & s_1^{(3)} = s_1^{(2)} + \lambda_3 s_0^{(2)} = \lambda_1 + \lambda_2 + \lambda_3 \\
 & & s_2^{(3)} = s_2^{(2)} + \lambda_3 s_1^{(2)} = \lambda_2\lambda_1 + \lambda_3(\lambda_1 + \lambda_2) \\
 & & s_3^{(3)} = s_3^{(2)} + \lambda_3 s_2^{(2)} = \lambda_3\lambda_2\lambda_1 \\
i=4: & s_1(\lambda) = s_1^{(4)} = s_1^{(3)} + \lambda_4 s_0^{(3)} = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 \\
 & s_2(\lambda) = s_2^{(4)} = s_2^{(3)} + \lambda_4 s_1^{(3)} = \lambda_2\lambda_1 + \lambda_3(\lambda_1 + \lambda_2) + \lambda_4(\lambda_1 + \lambda_2 + \lambda_3) \\
 & s_3(\lambda) = s_3^{(4)} = s_3^{(3)} + \lambda_4 s_2^{(3)} = \lambda_3\lambda_2\lambda_1 + \lambda_4\left[\lambda_2\lambda_1 + \lambda_3(\lambda_1 + \lambda_2)\right] \\
 & s_4(\lambda) = s_4^{(4)} = s_4^{(3)} + \lambda_4 s_3^{(3)} = \lambda_4\lambda_3\lambda_2\lambda_1.
\end{array}
$$

4. Roundoff error bounds. We derive error bounds for Algorithm 1. For elementary symmetric functions with exact inputs, we derive roundoff error bounds in section 4.1 and running error bounds in section 4.2. Then we apply these bounds to characteristic polynomials. In section 4.3 we derive roundoff and running error bounds for characteristic polynomials computed from exact eigenvalues, and in section 4.4 we extend these bounds to characteristic polynomials computed from perturbed eigenvalues.

We assume that the inputs λ are real. Error bounds for complex values are derived in [12, section 5.2.5]. The quantities $\hat{s}_k^{(i)}$ denote the values $s_k^{(i)}$ computed by Algorithm 1 in floating point arithmetic as follows:

$$\hat{s}_k^{(i)} = \mathrm{fl}\left[\hat{s}_k^{(i-1)} + \mathrm{fl}\left[\lambda_i \hat{s}_{k-1}^{(i-1)}\right]\right], \qquad 2 \le i \le n, \quad 1 \le k \le n-1.$$

The output of Algorithm 1 is $\hat{s}_k(\lambda) \equiv \hat{s}_k^{(n)}$.

Assumptions 4.1.

1. The elements of $\lambda = (\lambda_1 \quad \dots \quad \lambda_n)$ are normalized real floating point numbers.
2. Algorithm 1 computes the following quantities exactly:
$$\hat{s}_0^{(l)} = s_0^{(l)}, \quad 1 \le l \le n-1, \qquad \hat{s}_k^{(l)} = s_k^{(l)}, \quad k > l, \qquad \hat{s}_1^{(1)} = s_1^{(1)}.$$
3. The operations in Algorithm 1 do not cause underflow or overflow.

4. The problem size is limited by $nu < 1$, where u denotes the unit roundoff.
5. Standard model for real floating point arithmetic [7, section 2.2]:
   If $\mathrm{op} \in \{+, -, \times, /\}$ and x and y are real normalized floating point numbers so that $x \,\mathrm{op}\, y$ does not underflow or overflow, then

$$\mathrm{fl}[x \,\mathrm{op}\, y] = (x \,\mathrm{op}\, y)(1+\delta), \qquad \text{where} \qquad |\delta| \le u, \tag{4.1}$$

   and

$$\mathrm{fl}[x \,\mathrm{op}\, y] = \frac{x \,\mathrm{op}\, y}{1+\delta}, \qquad \text{where} \qquad |\delta| \le u. \tag{4.2}$$

The following relations are required for the error bounds.

LEMMA 4.1 (Lemmas 3.1 and 3.3 in [7]). *Let δ_i and ρ_i be real numbers, $1 \le i \le n$, with $|\delta_i| \le u$ and $\rho_i = \pm 1$. If $nu < 1$, then*

1. $\prod_{i=1}^{n}(1+\delta_i)^{\rho_i} = 1 + \theta_n$, *where*
$$|\theta_n| \le \gamma_n \equiv \frac{nu}{1-nu},$$
2. $(1+\theta_j)(1+\theta_k) = 1 + \theta_{j+k}$,
3. $\gamma_j + \gamma_k + \gamma_j\gamma_k \le \gamma_{j+k}$.

4.1. Roundoff error bounds for elementary symmetric functions. We derive roundoff error bounds for the elementary symmetric functions $s_k(\lambda)$, assuming the real inputs λ are known exactly.

LEMMA 4.2 (error expansions). *If λ is real, the $\hat{s}_k^{(i)}$ are computed by Algorithm* 1, *and Assumptions* 4.1 *hold then*

$$\begin{aligned}\hat{s}_k(\lambda) \equiv \hat{s}_k^{(n)} &= \mathrm{fl}\left[\hat{s}_k^{(n-1)} + \mathrm{fl}\left[\lambda_n \hat{s}_{k-1}^{(n-1)}\right]\right] \\ &= \sum_{1 \le i_1 < \cdots < i_k \le n} \lambda_{i_1} \cdots \lambda_{i_k} \left(1 + \theta_t^{(i_1 \ldots i_k)}\right),\end{aligned}$$

where for each index set $(i_1 \ldots i_k)$ the value t is an integer with $1 \le t \le 2n$.

Proof. The proof proceeds by induction on the number i of inputs $\lambda_1, \ldots, \lambda_i$. Since there are no floating point operations for $i = 1$, we prove the induction basis for $i = 2$. The model (4.1) implies

$$\hat{s}_1^{(2)} = (\lambda_1 + \lambda_2)(1+\delta_1) = (\lambda_1 + \lambda_2)(1+\theta_1),$$

where $|\delta_1| \le u$, and we used part 1 of Lemma 4.1 to set $\theta_1 \equiv \delta_1$. Similarly,

$$\hat{s}_2^{(2)} = \lambda_1\lambda_2(1+\delta_2) = \lambda_1\lambda_2\left(1 + \theta_1^{(12)}\right),$$

where $|\delta_2| \le u$ and $\theta_1^{(12)} \equiv \delta_2$. Hence the statement holds for $i = 2$.

Suppose that the statement also holds for $i = n-1$. We prove the induction step for $i = n$ input values. From

$$\hat{s}_k^{(n)} = \mathrm{fl}\left[\hat{s}_k^{(n-1)} + \mathrm{fl}\left[\lambda_n \hat{s}_{k-1}^{(n-1)}\right]\right]$$

and (4.1) follows that

$$\begin{aligned}\hat{s}_k^{(n)} &= \hat{s}_k^{(n-1)}(1+\delta_3) + \lambda_n \hat{s}_{k-1}^{(n-1)}(1+\delta_3)(1+\delta_4) \\ &= \hat{s}_k^{(n-1)}(1+\theta_1) + \lambda_n \hat{s}_{k-1}^{(n-1)}(1+\theta_2).\end{aligned}$$

The induction hypothesis implies

$$\hat{s}_k^{(n-1)} = \sum_{1 \le i_1 < \cdots < i_k \le n-1} \lambda_{i_1} \cdots \lambda_{i_k} \left(1 + \theta_{t_1}^{(i_1 \ldots i_k)}\right),$$

where for each $(i_1 \ldots i_k)$ the value t_1 is an integer with $1 \le t_1 \le 2n-2$, as well as

$$\hat{s}_{k-1}^{(n-1)} = \sum_{1 \le i_1 < \cdots < i_{k-1} \le n-1} \lambda_{i_1} \cdots \lambda_{i_{k-1}} \left(1 + \theta_{t_2}^{(i_1 \ldots i_{k-1})}\right),$$

where for each $(i_1 \ldots i_{k-1})$ the value t_2 is an integer with $1 \le t_2 \le 2n-2$. Substituting these two expressions into the one for $\hat{s}_k^{(n)}$ gives

$$\begin{aligned}\hat{s}_k^{(n)} = & \sum_{1 \le i_1 < \cdots < i_k \le n-1} \lambda_{i_1} \cdots \lambda_{i_k} \left(1 + \theta_{t_1}^{(i_1 \ldots i_k)}\right)(1 + \theta_1) \\ & + \lambda_n \sum_{1 \le i_1 < \cdots < i_{k-1} \le n-1} \lambda_{i_1} \cdots \lambda_{i_{k-1}} \left(1 + \theta_{t_2}^{(i_1 \ldots i_{k-1})}\right)(1 + \theta_2).\end{aligned}$$

Finally, applying part 2 of Lemma 4.1 to both sums gives

$$\hat{s}_k^{(n)} = \sum_{1 \le i_1 < \cdots < i_k \le n} \lambda_{i_1} \cdots \lambda_{i_k} \left(1 + \theta_t^{(i_1 \ldots i_k)}\right),$$

where for each $(i_1 \ldots i_k)$ the value t is an integer with $1 \le t \le 2n$. ∎

We use the above expressions for the errors to derive roundoff error bounds for elementary symmetric functions computed by Algorithm 1 from exact inputs.

THEOREM 4.3 (worst case roundoff error bounds). *If λ is real, the $\hat{s}_k(\lambda)$ are computed by Algorithm* 1*, Assumptions* 4.1 *hold, and* $2nu < 1$*, then*

$$|\hat{s}_k(\lambda) - s_k(\lambda)| \le \gamma_{2n}\, s_k(|\lambda|) \le \frac{2nu}{1-2nu}\, s_k(|\lambda|), \qquad 1 \le k \le n-1,$$

and

$$|\hat{s}_n(\lambda) - s_n(\lambda)| \le \gamma_{n-1}\, |s_n(\lambda)| \le \frac{(n-1)u}{1-(n-1)u}\, |s_n(\lambda)|.$$

Proof. Lemma 4.2 implies for $1 \le k \le n-1$

$$\begin{aligned}\hat{s}_k(\lambda) &= \sum_{1 \le i_1 < \cdots < i_k \le n} \lambda_{i_1} \cdots \lambda_{i_k} \left(1 + \theta_t^{(i_1 \ldots i_k)}\right) \\ &= s_k(\lambda) + \sum_{1 \le i_1 < \cdots < i_k \le n} \theta_t^{(i_1 \ldots i_k)} \lambda_{i_1} \cdots \lambda_{i_k},\end{aligned}$$

where for each $(i_1 \ldots i_k)$ the value t is an integer with $1 \le t \le 2n$. Applying the triangle inequality to every summand gives

$$|\hat{s}_k(\lambda) - s_k(\lambda)| \le \sum_{1 \le i_1 < \cdots < i_k \le n} |\theta_t^{(i_1 \ldots i_k)}|\, |\lambda_{i_1}| \cdots |\lambda_{i_k}|.$$

According to part 1 of Lemma 4.1, we can bound every $|\theta_t^{(i_1 \ldots i_k)}|$ by γ_{2n} to get the desired result. The bound for $\hat{s}_n(\lambda)$ follows from the fact that Algorithm 1 computes $s_n(\lambda)$ as the product $\lambda_1 \cdots \lambda_n$. ∎

Remark 4.1. *Algorithm* 1 *is forward stable.*

Comparing Theorem 4.3 *with Corollary* 2.10 *shows that with* $\epsilon_{rel} = u$ *the roundoff error bounds of the elementary symmetric functions* $\hat{s}_k(\lambda)$ *computed by Algorithm* 1 *are close to the perturbation bounds for* $s_k(\hat{\lambda})$, $1 \leq k \leq n$.

Theorem 4.3 also implies that Algorithm 1 computes elementary symmetric functions with positive inputs to high relative accuracy.

COROLLARY 4.4 (positive inputs). *If* $\lambda > 0$, *the* $\hat{s}_k(\lambda)$ *are computed by Algorithm* 1, *Assumptions* 4.1 *hold, and* $2nu < 1$, *then*

$$\frac{|\hat{s}_k(\lambda) - s_k(\lambda)|}{s_k(\lambda)} \leq \gamma_{2n} \leq \frac{2nu}{1-2nu}, \qquad 1 \leq k \leq n.$$

4.2. Running error bounds for elementary symmetric functions. The perturbation bounds of Algorithm 1 in Theorem 4.3 are worst case bounds that do not depend on actual rounding errors committed during the computations and that do not take into account possible cancellation in intermediate quantities. We derive sharper running error bounds for Algorithm 1. The idea is to compute error bounds from computed values at every step of the recursion so that we can take advantage of cancellation that might occur in intermediate quantities. There are, of course, rounding errors in the computation of the running error bounds, but their effect is negligible [7, section 3.3].

We denote the error in the computed elementary function by $e_k^{(i)}$; that is,

$$\hat{s}_k^{(i)} = s_k^{(i)} + e_k^{(i)}, \qquad 2 \leq i \leq n, \quad 1 \leq k \leq n-1.$$

We present recursions for $|e_k^{(i)}|$ that can be used in conjunction with Algorithm 1 to bound the final error $|e_k^{(n)}|$ in $\hat{s}_k(\lambda)$ from intermediate quantities. We start with a recursion for bounding the error associated with the first elementary symmetric function $\hat{s}_1(\lambda)$.

THEOREM 4.5 (running error bounds for $\hat{s}_1^{(i)}$). *If* λ *is real, the* $\hat{s}_1^{(i)}$ *are computed by Algorithm* 1, *and Assumptions* 4.1 *hold, then*

$$\left|e_1^{(i)}\right| \leq \left|e_1^{(i-1)}\right| + u \left|\hat{s}_1^{(i)}\right|, \qquad 2 \leq i \leq n.$$

Proof. According to the model (4.2), we can write

$$\hat{s}_1^{(i)} = \frac{\hat{s}_1^{(i-1)} + \lambda_i}{1+\delta^{(i)}}, \qquad 2 \leq i \leq n,$$

where $|\delta^{(i)}| \leq u$. Hence

$$\hat{s}_1^{(i-1)} + \lambda_i = (1+\delta^{(i)})\, \hat{s}_1^{(i)} = \hat{s}_1^{(i)} + \delta^{(i)} \hat{s}_1^{(i)}.$$

Representing $\hat{s}_1^{(i-1)}$ and $\hat{s}_1^{(i)}$ in terms of their errors and simplifying gives

$$e_1^{(i)} = e_1^{(i-1)} - \delta^{(i)} \hat{s}_1^{(i)}.$$

Thus $|e_1^{(i)}| \leq |e_1^{(i-1)}| + u\,|\hat{s}_1^{(i)}|$. ∎

Now we derive bounds for the errors associated with the remaining elementary symmetric functions.

THEOREM 4.6 (running error bounds for all other $\hat{s}_k^{(i)}$). *If λ is real, the $\hat{s}_k^{(i)}$ are computed by Algorithm* 1*, and Assumptions* 4.1 *hold, then*

$$\left|e_k^{(k)}\right| \leq \left|\lambda_k e_{k-1}^{(k-1)}\right| + u\ \left|\hat{s}_k^{(k)}\right|, \qquad 2 \leq k \leq n,$$

and

$$\left|e_k^{(i)}\right| \leq \left|e_k^{(i-1)}\right| + \left|\lambda_i e_{k-1}^{(i-1)}\right| + u\ \left(\left|\lambda_i \hat{s}_{k-1}^{(i-1)}\right| + \left|\hat{s}_k^{(i)}\right|\right), \qquad k < i \leq n.$$

Proof. Since $s_k^{(i)} = 0$ for $k > i$, Algorithm 1 starts accumulating errors in $\hat{s}_k^{(i)}$ at step $i = k$ so that $\hat{s}_k^{(k)} = \text{fl}\left[\lambda_k \hat{s}_{k-1}^{(k-1)}\right]$. According to the model (4.2), we can write $(1+\delta)\hat{s}_k^{(k)} = \lambda_k \hat{s}_{k-1}^{(k-1)}$, where $|\delta| \leq u$. Expressing $\hat{s}_{k-1}^{(k-1)}$ and $\hat{s}_k^{(k)}$ in terms of their errors and simplifying yields $e_k^{(k)} = \lambda_k e_{k-1}^{(k-1)} - \delta \hat{s}_k^{(k)}$. The triangle inequality gives the desired running error bound for $\hat{s}_k^{(k)}$.

For $k < i \leq n$, we use (4.1) and (4.2) to write

$$(1+\epsilon^{(i)})\hat{s}_k^{(i)} = \hat{s}_k^{(i-1)} + \lambda_i \hat{s}_{k-1}^{(i-1)}(1+\delta^{(i)}),$$

where $|\epsilon^{(i)}|, |\delta^{(i)}| \leq u$. Expressing $\hat{s}_{k-1}^{(i-1)}$ and $\hat{s}_k^{(i)}$ in terms of their errors and simplifying produces

$$e_k^{(i)} = e_k^{(i-1)} + \lambda_i e_{k-1}^{(i-1)} + \delta^{(i)} \lambda_i \hat{s}_{k-1}^{(i-1)} - \epsilon^{(i)} \hat{s}_k^{(i)}.$$

At last, the triangle inequality gives the desired bound. ∎

4.3. Roundoff error bounds for characteristic polynomials computed from exact eigenvalues. We apply the bounds for elementary symmetric functions in sections 4.1 and 4.2 to characteristic polynomials that are computed with Algorithm 1 from exact eigenvalues.

THEOREM 4.7 (roundoff error bounds). *If λ is real,* $\text{fl}[c_k] = (-1)^k\ \hat{s}_k(\lambda)$*, where the $\hat{s}_k(\lambda)$ are computed by Algorithm* 1*, Assumptions* 4.1 *hold, and* $2nu < 1$*, then*

$$|\,\text{fl}[c_k] - c_k| \leq \gamma_{2n}\ s_k(|\lambda|), \qquad 1 \leq k \leq n.$$

If, in addition, $\lambda > 0$, then

$$|\,\text{fl}[c_k] - c_k| \leq \gamma_{2n}\ |c_k|, \qquad 1 \leq k \leq n.$$

Proof. The first bound follows from Lemma 2.2 and Theorem 4.3, and the second one follows from Lemma 2.2 and Corollary 4.4. ∎

The following absolute error bounds ρ_k represent running error bounds for characteristic polynomials derived from Theorems 4.5 and 4.6. The quantities $\rho_k^{(i)}$ below are bounds for the intermediate absolute errors $|\,\text{fl}[c_k^{(i)}] - c_k^{(i)}|$.

THEOREM 4.8 (running error bounds). *If λ is real,* $\text{fl}[c_k] = (-1)^k\ \hat{s}_k(\lambda)$*, where the $\hat{s}_k(\lambda)$ are computed by Algorithm* 1*, and Assumptions* 4.1 *hold, then*

$$|\,\text{fl}[c_k] - c_k| \leq \rho_k, \qquad 1 \leq k \leq n,$$

where

$$\begin{aligned}
\rho_k^{(1)} &= 0, \qquad 1 \le k \le n,\\
\rho_1^{(i)} &= \rho_1^{(i-1)} + u\,\left|\hat{s}_1^{(i)}\right|, \qquad 2 \le i \le n,\\
\rho_k^{(k)} &= |\lambda_k|\,\rho_{k-1}^{(k-1)} + u\,\left|\hat{s}_k^{(k)}\right|, \qquad 2 \le k \le n,\\
\rho_k^{(i)} &= \rho_k^{(i-1)} + |\lambda_i|\,\rho_{k-1}^{(i-1)} + u\,\left(\left|\lambda_i \hat{s}_{k-1}^{(i-1)}\right| + \left|\hat{s}_k^{(i)}\right|\right), \qquad k < i \le n,\\
\rho_k &= \rho_k^{(n)}, \qquad 1 \le k \le n.
\end{aligned}$$

4.4. Roundoff error bounds for characteristic polynomials computed from perturbed eigenvalues. At last we derive roundoff error bounds for characteristic polynomials when Algorithm 1 is applied to computed eigenvalues. We do this by combining the roundoff error bounds for exact eigenvalues in section 4.3 with the perturbation bounds in section 2.3.

We start with eigenvalues that have absolute errors. That is, $\tilde{\lambda}_i \equiv \lambda_i + \epsilon_i$ are the eigenvalues of a matrix whose characteristic polynomial is $z^n + \tilde{c}_1 z^{n-1} + \cdots + \tilde{c}_{n-1} z + \tilde{c}_n$, and $\epsilon_{abs} \equiv \max_{1\le i\le n} |\epsilon_i|$. We denote by $\mathrm{fl}[\tilde{c}_k]$ the coefficients computed by Algorithm 1 in floating point arithmetic from the perturbed eigenvalues $\tilde{\lambda}_i$.

THEOREM 4.9 (eigenvalues with absolute errors). *If $\tilde{\lambda}_i$ are real, Assumptions* 4.1 *hold, $2nu < 1$, and $\epsilon_{abs} < 1$, then*

$$|\,\mathrm{fl}[\tilde{c}_k] - c_k| \le (n-k+1) s_{k-1}(|\lambda|)\,(1+\gamma_{2n})\epsilon_{abs} + s_k(|\lambda|)\,\gamma_{2n} + \mathcal{O}\left(\epsilon_{abs}^2\right).$$

If, in addition, $\lambda > 0$ and $\tilde{\lambda} > 0$, then

$$|\,\mathrm{fl}[\tilde{c}_k] - c_k| \le (n-k+1)|c_{k-1}|\,(1+\gamma_{2n})\epsilon_{abs} + |c_k|\,\gamma_{2n} + \mathcal{O}\left(\epsilon_{abs}^2\right).$$

Proof. The triangle inequality implies

$$|\,\mathrm{fl}[\tilde{c}_k] - c_k| \le |\,\mathrm{fl}[\tilde{c}_k] - \tilde{c}_k| + |\tilde{c}_k - c_k|, \qquad 1 \le k \le n.$$

Applying the perturbation bound in Theorem 2.12 to the second summand gives

$$|\tilde{c}_k - c_k| \le (n-k+1) s_{k-1}(|\lambda|)\,\epsilon_{abs} + \mathcal{O}\left(\epsilon_{abs}^2\right).$$

To the first summand we apply the roundoff error bound in Theorem 4.7,

$$|\,\mathrm{fl}[\tilde{c}_k] - \tilde{c}_k| \le \gamma_{2n}\, s_k(|\tilde{\lambda}|), \qquad 1 \le k \le n,$$

and then bound $s_k(|\tilde{\lambda}|)$ with the perturbation bound (2.2)

$$s_k(|\tilde{\lambda}|) \le s_k(|\lambda|) + (n-k+1) s_{k-1}(|\lambda|)\,\epsilon_{abs} + \mathcal{O}\left(\epsilon_{abs}^2\right), \qquad 1 \le k \le n. \qquad \square$$

Theorem 4.9 implies that floating point arithmetic introduces a second amplification factor, $s_k(|\lambda|)$, in addition to $s_{k-1}(|\lambda|)$.

Next we consider eigenvalues with relative errors. That is, $\hat{\lambda}_i \equiv \lambda_i(1+\epsilon_i)$ are the eigenvalues of a matrix whose characteristic polynomial is $z^n + \hat{c}_1 z^{n-1} + \cdots + \hat{c}_{n-1} z + \hat{c}_n$, and $\epsilon_{rel} \equiv \max_{1\le i\le n} |\epsilon_i|$. As before, we denote by $\mathrm{fl}[\hat{c}_k]$ the coefficients computed by Algorithm 1 in floating point arithmetic from the perturbed eigenvalues $\hat{\lambda}_i$.

THEOREM 4.10 (eigenvalues with relative errors). *If $\hat{\lambda}_i$ are real, Assumptions* 4.1 *hold, $2nu < 1$, $\epsilon_{rel} < 1$, and $c_k \neq 0$, then*

$$\frac{|\,\mathrm{fl}[\hat{c}_k] - c_k|}{|c_k|} \leq \frac{s_k(|\lambda|)}{|c_k|}\,\frac{\gamma_{2n} + k\epsilon_{rel}}{1 - k\epsilon_{rel}}, \qquad 1 \leq k \leq n.$$

If, in addition, $\lambda > 0$, then

$$\frac{|\,\mathrm{fl}[\hat{c}_k] - c_k|}{|c_k|} \leq \frac{\gamma_{2n} + k\epsilon_{rel}}{1 - k\epsilon_{rel}}, \qquad 1 \leq k \leq n.$$

Proof. As in the proof of Theorem 4.9, we start with the triangle inequality,

$$|\,\mathrm{fl}[\hat{c}_k] - c_k| \leq |\,\mathrm{fl}[\hat{c}_k] - \hat{c}_k| + |\hat{c}_k - c_k|, \qquad 1 \leq k \leq n.$$

Applying the perturbation bound in Theorem 2.13 to the second summand gives

$$|\hat{c}_k - c_k| \leq \frac{k\epsilon_{rel}}{1 - k\epsilon_{rel}}\, s_k(|\lambda|).$$

To the first summand we apply the roundoff error bound in Theorem 4.7,

$$|\,\mathrm{fl}[\hat{c}_k] - \hat{c}_k| \leq \gamma_{2n}\, s_k(|\hat{\lambda}|), \qquad 1 \leq k \leq n,$$

and then bound $s_k(|\hat{\lambda}|)$ with the perturbation bound from Corollary 2.10,

$$s_k(|\hat{\lambda}|) \leq s_k(|\lambda|) + \frac{k\epsilon_{rel}}{1 - k\epsilon_{rel}}\, s_k(|\lambda|) = \frac{s_k(|\lambda|)}{1 - k\epsilon_{rel}}, \qquad 1 \leq k \leq n. \qquad \square$$

Theorem 4.10 implies that, in floating point arithmetic, relative perturbations in the eigenvalues are amplified by $s_k(|\lambda|)/|c_k|$.

5. Numerical experiments. We apply the bounds in section 4.4 and the running error bounds in Theorem 4.8 to estimate the accuracy of characteristic polynomials computed with the `poly` function in MATLAB. Numerical experiments are presented for several classes of matrices: Forsythe (section 5.1), symmetric positive definite (section 5.2), symmetric indefinite (section 5.3), symmetric indefinite tridiagonal Toeplitz (section 5.4), companion (section 5.5), and Frank (section 5.6) matrices.

The `poly` function first computes the eigenvalues λ_j with the `eig` function and then determines the coefficients as follows:

```
c = [1 zeros(1,n)]
for j = 1:n
c(2:(j+1)) = c(2:(j+1)) - λ_j.*c(1:j)
end
```

Thus `poly` is the same as Algorithm 1, except that it computes the coefficients $c_k =$`c(k-1)` directly, rather than computing the elementary symmetric functions $s_k(\lambda)$ first and then multiplying them by $(-1)^k$; see Lemma 2.2. We denote the coefficients computed with `poly` function by c_k^{poly} and estimate their error with the three types of bounds below.

Absolute bounds from Theorem 4.9. We denote the absolute error in the coefficients by

$$\eta_k \equiv |c_k^{poly} - c_k|, \qquad 1 \leq k \leq n.$$

If the exact quantities c_k are not available then we estimate η_k from

$$\eta_k^A \equiv (n-k+1)s_{k-1}(|\lambda|)\,(1+\gamma_{2n})\epsilon_{abs} + s_k(|\lambda|)\,\gamma_{2n}, \tag{5.1}$$

where ϵ_{abs} is the maximal absolute error in the computed eigenvalues. If $\lambda > 0$ then

$$\eta_k^A \equiv (n-k+1)|c_{k-1}|\,(1+\gamma_{2n})\epsilon_{abs} + |c_k|\,\gamma_{2n}. \tag{5.2}$$

Relative bounds from Theorem 4.10. We estimate the relative error $|c_k^{poly} - c_k|/|c_k|$ from

$$\eta_k^R \equiv \frac{s_k(|\lambda|)}{|c_k|}\,\frac{\gamma_{2n} + k\epsilon_{rel}}{1 - k\epsilon_{rel}}, \tag{5.3}$$

where ϵ_{rel} is the relative error in the computed eigenvalues. If $\lambda > 0$ then

$$\eta_k^R \equiv \frac{\gamma_{2n} + k\epsilon_{rel}}{1 - k\epsilon_{rel}}. \tag{5.4}$$

Running error bounds from Theorem 4.8. We estimate the absolute error η_k from the running error bounds ρ_k.

The machine precision is IEEE double precision $u \approx 1.1 \times 10^{-16}$. To distinguish the characteristic polynomials of different matrices, we use $c_k(X)$ to denote the kth coefficient of the characteristic polynomial of the matrix X.

5.1. The Forsythe matrix. This example illustrates that the coefficients of a characteristic polynomial can be extremely sensitive to errors in the computed eigenvalues and that the absolute error bounds (5.1) are able to capture this.

Our version of the Forsythe matrix is a perturbed Jordan block of order $n = 200$ with zero eigenvalues and a perturbation 10^{-10} in the $(n, 1)$ entry:

$$F = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ \nu & & & 0 \end{pmatrix}, \quad \text{where} \quad \nu = 10^{-10}.$$

The characteristic polynomial of F is $p(\lambda) = \lambda^n - \nu$ so that all coefficients except for $c_{200}(F) = -\nu$ are zero.

Figure 5.1 shows, however, that many coefficients computed with the `poly` function in MATLAB are not only nonzero but have enormous magnitudes. The largest coefficient has magnitude 10^{28}. To understand why this tremendous loss of accuracy occurs, note that the eigenvalues of F are complex numbers of the form [6, Example 5.22]:

$$\lambda_j = \sqrt[n]{\nu}\,\exp\left(\frac{2j\pi\imath}{n}\right), \quad \text{where} \quad \imath^2 = -1, \qquad 1 \le j \le n.$$

From these exact expressions, we can compute the largest absolute error $\epsilon_{abs} \approx 1.78$ in the eigenvalues (the eigenvalues were ordered according to the real part). Figure 5.1 illustrates that the absolute bounds η_K^A from (5.1) reflect the huge absolute errors and capture the extreme ill-conditioning of the coefficients.

One might think that the large errors in the c_k could be in part due to the high multiplicity of the eigenvalues. The next example shows that this is not the case.

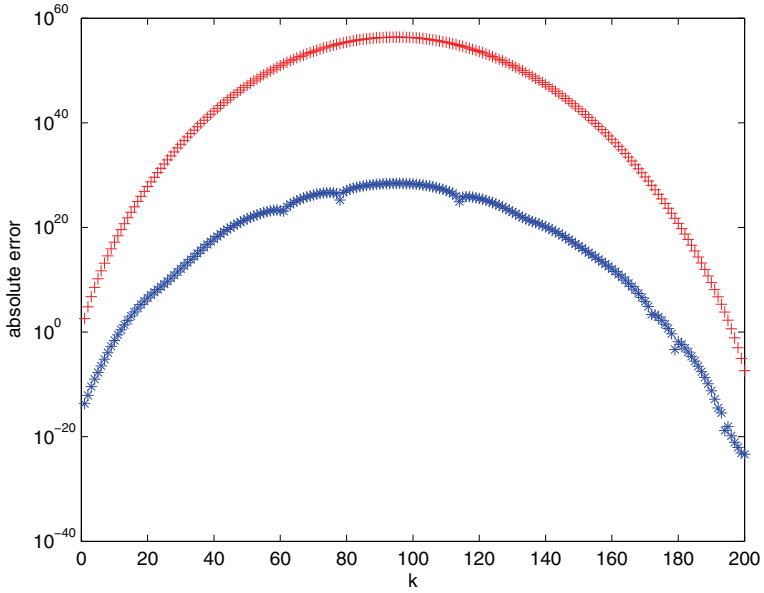

FIG. 5.1. *The Forsythe matrix F. Lower ($*$) curve: Absolute values of the computed coefficients $c_k^{poly}(F)$. Note that the exact coefficients $c_k(F)$ are zero, $1 \le k \le 199$. Upper ($+$) curve: Absolute bounds η_k^A from* (5.1) *with $\epsilon_{abs} \approx 1.78$.*

5.2. Symmetric positive definite matrix. This example illustrates that the absolute error bounds (5.2) and the running error bounds from Theorem 4.8 can predict the absolute error in the coefficients well, and that the `poly` function in MATLAB can compute the characteristic polynomial of a real symmetric positive definite matrix to high absolute accuracy.

The $n \times n$ symmetric positive definite matrix is

$$H = Q \begin{pmatrix} I_{n/2} & \\ & 2I_{n/2} \end{pmatrix} Q^*,$$

where Q is a random orthogonal matrix from `[Q, R] = qr(rand(n, n))`. The computed eigenvalues have absolute accuracy $\epsilon_{abs} \approx 4 \cdot 10^{-15}$. We obtained the exact coefficients $c_k(H)$ with `sym2poly(poly(sym(H)))` from the Symbolic Math Toolbox in MATLAB.

Figure 5.2 illustrates for $n = 200$ that the absolute bound (5.2) captures the absolute error in the computed coefficients $c_k^{poly}(H)$ very well, as do the running error bounds ρ_k from Theorem 4.8.

In our experiments we observed that MATLAB often computes eigenvalues of Hermitian positive definite matrices to high relative accuracy. Therefore, we also plotted relative bounds for $n = 200$. Figure 5.3 illustrates that `poly` computes the coefficients $c_k^{poly}(H)$ to high relative accuracy. The relative bound (5.4) with $\epsilon_{rel} \approx 4 \cdot 10^{-15}$ captures the low relative error, but becomes more pessimistic for coefficients c_k with larger k.

5.3. Symmetric indefinite matrix. This example illustrates that the absolute error bounds (5.2) and the running error bounds from Theorem 4.8 also predict the absolute error well when the matrix is indefinite. The enormous loss of accuracy here is

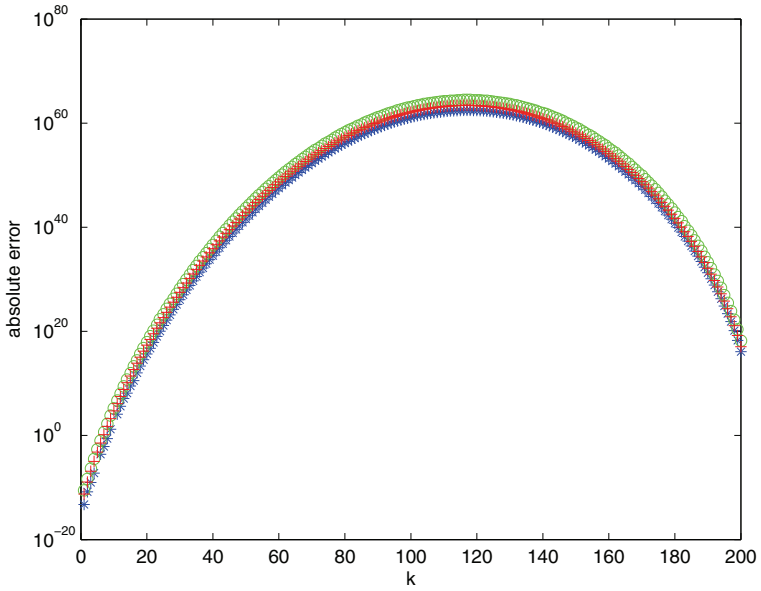

FIG. 5.2. *Symmetric positive definite matrix H. Lower $(*)$ curve: Absolute errors η_k. Middle $(+)$ curve: Absolute bound η_k^A from* (5.2) *with $\epsilon_{abs} \approx 4 \cdot 10^{-15}$. Upper (o) curve: Running error bounds ρ_k from Theorem* 4.8.

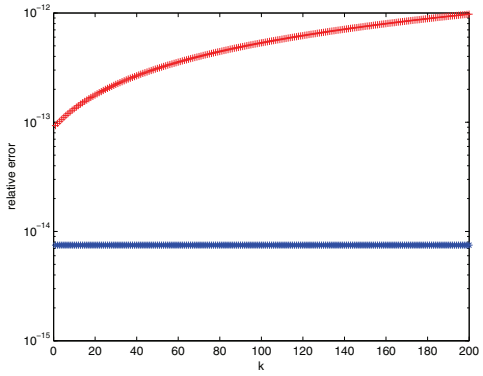

FIG. 5.3. *Symmetric positive definite matrix H. Lower $(*)$ line: Relative errors $|c_k(H) - c_k^{poly}(H)|/|c_k(H)|$. Upper $(+)$ curve: Relative bound η_k^R from* (5.4) *with $\epsilon_{rel} \approx 4 \cdot 10^{-15}$.*

due to the ill-conditioning of the coefficients, because the eigenvalues were computed to high absolute and relative accuracy.

Similar to the positive definite matrix in section 5.2, we define the $n \times n$ symmetric indefinite definite matrix as

$$J = Q \begin{pmatrix} I_{n/2} & \\ & -I_{n/2} \end{pmatrix} Q^*,$$

where Q is a random orthogonal matrix from `[Q, R] = qr(rand(n, n))`. As before, the exact coefficients $c_k(J)$ were determined with `sym2poly(poly(sym(J)))` for $n = 200$. Every other coefficient is zero so that $c_k(J) = 0$ for k odd. The nonzero

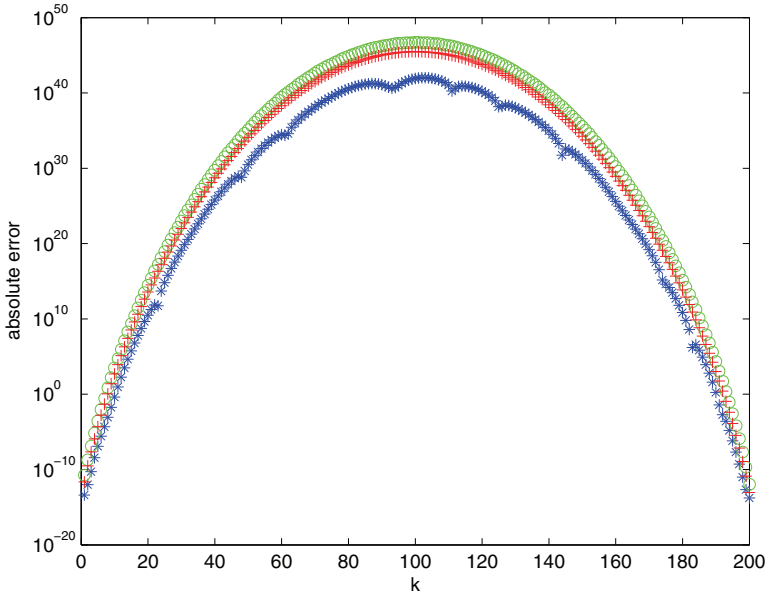

FIG. 5.4. *Symmetric indefinite matrix J. Lower ($*$) curve: Absolute errors η_k. Middle ($+$) curve: Absolute bound η_k^A from* (5.2) *with $\epsilon_{abs} \approx 4 \cdot 10^{-15}$. Upper ($o$) curve: Running error bounds ρ_k from Theorem* 4.8.

coefficients oscillate in sign and have widely varying magnitudes, with $c_{100}(J) \geq 10^{29}$ and $c_{200}(J) = 1$. Figure 5.4 illustrates that the absolute bound (5.2) and the running error bounds ρ_k from Theorem 4.8 capture the absolute error in the computed coefficients $c_k^{poly}(J)$ very well.

MATLAB computes the eigenvalues of J to the same high accuracy as the eigenvalues of the positive definite matrix T: $\epsilon_{abs} = \epsilon_{rel} \approx 4 \cdot 10^{-15}$, perhaps because all eigenvalues of J have the same magnitude. Hence the loss of accuracy in the coefficients $c_k(J)$ can only be due to their severe ill-conditioning.

5.4. Symmetric indefinite Toeplitz matrix. This example illustrates that for a symmetric indefinite matrix the running error bounds in Theorem 4.8 can be tighter than the absolute bound (5.1).

The matrix is an $n \times n$ symmetric indefinite tridiagonal Toeplitz matrix

$$T = \begin{pmatrix} 0 & 100 & & \\ 100 & \ddots & \ddots & \\ & \ddots & 0 & 100 \\ & & 100 & 0 \end{pmatrix}.$$

For $n = 100$ we obtained the exact coefficients $c_k(T)$ with `sym2poly(poly(sym(T)))` from the Symbolic Math Toolbox in MATLAB. We used the exact expressions for the eigenvalues [7, section 28.5] $\lambda_j = 200 \cos \frac{j\pi}{n+1}$, $1 \leq j \leq n$, to estimate the maximal absolute error in the eigenvalues computed by `eig` and obtained $\epsilon_{abs} \approx 10^{-13}$. To compare the tightness of the worst case bounds η_k^A and the running error bound ρ_k from Theorem 4.8, we plot the relative differences between the exact error and the bounds $|\eta_k - \eta_k^A|/|\eta_k|$ and $|\eta_k - \rho_k|/|\eta_k|$. Figure 5.5 illustrates that the relative

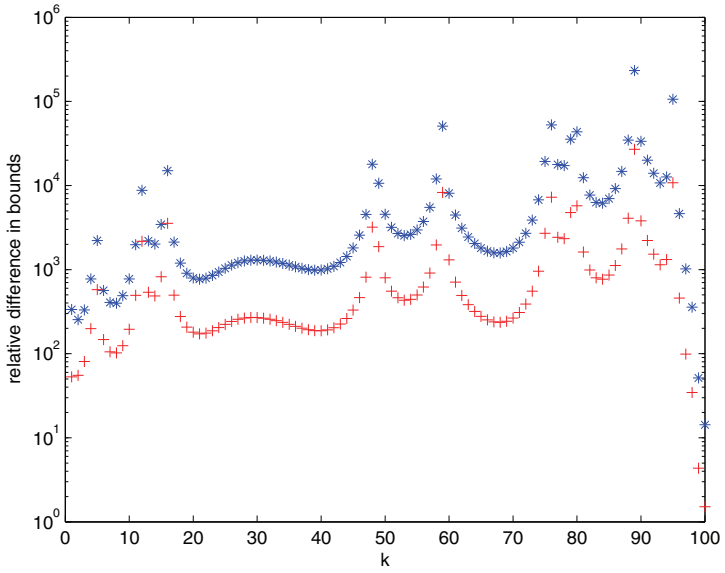

FIG. 5.5. *Symmetric indefinite Toeplitz matrix T. Upper $(*)$ curve: Relative tightness $|\eta_k - \eta_k^A|/|\eta_k|$ of the absolute bound η_k^A from (5.1) with $\epsilon_{abs} \approx 10^{-13}$ compared to the absolute error η_k. Lower $(+)$ curve: Relative tightness $|\eta_k - \rho_k|/|\eta_k|$ of the running error bound ρ_k from Theorem 4.8.*

tightness of the running error bounds from Theorem 4.8 can be several magnitudes better than that of the absolute bound (5.1).

5.5. Companion matrix. One type of matrix for which the coefficients of a characteristic polynomial are easy to determine is a companion matrix, because the first row of a companion matrix contains the coefficients of the characteristic polynomial. We chose a companion matrix with coefficients that increase in magnitude,

$$C = \begin{pmatrix} -c_1 & \cdots & -c_n \\ 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix}, \qquad \text{where} \qquad c_k = 2^k. \tag{5.5}$$

Figure 5.6 shows that the absolute bounds (5.1) with a very optimistic eigenvalue accuracy of $\epsilon_{abs} = u$ track the error distribution well.

5.6. The Frank matrix. This example shows that absolute bounds based on first order perturbation bounds, such as (5.1) and (5.2), may not be sufficient to bound the error in the polynomial coefficients, and that higher order effects need to be taken into account.

The Frank matrix U is an upper Hessenberg matrix with determinant 1 from the `gallery` command of test matrices in MATLAB. The coefficients of the characteristic polynomial appear in pairs, in the sense that $c_k(U) = c_{n-k}(U)$. The eigenvalues are positive and occur in reciprocal pairs.

For a Frank matrix of order $n = 20$, we used the Symbolic Math Toolbox in MATLAB to determine the exact coefficients $c_k(U)$ with the command `sym2poly(poly(sym(U)))` and the exact eigenvalues with `double(eig(sym(U)))`.

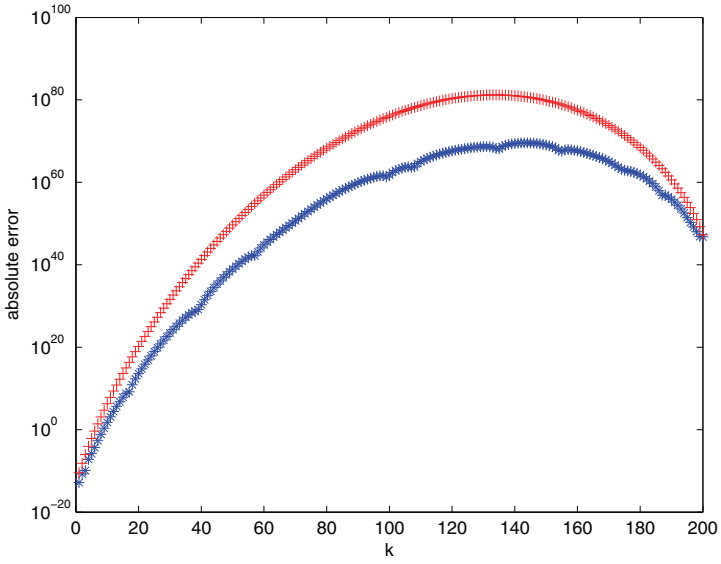

FIG. 5.6. *Companion matrix C. Lower $(*)$ curve: Absolute errors η_k. Upper $(+)$ curve: Absolute bound η_k^A from* (5.1) *with $\epsilon_{abs} = u$.*

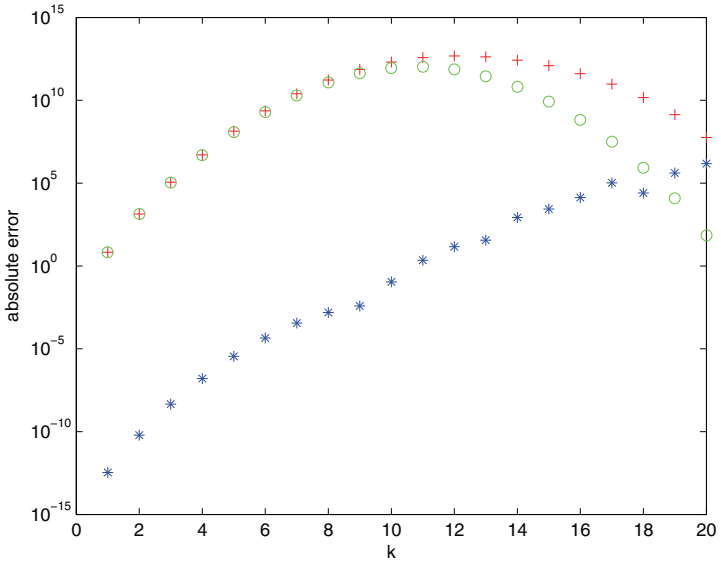

FIG. 5.7. *The Frank matrix U. Lower $(*)$ curve: Absolute errors η_k. Middle $(\circ)$ curve: Absolute bounds η_k^A from* (5.2) *with $\epsilon_{abs} = .3$. Upper $(+)$ curve: "Full order" bounds η_k^F from* (5.6) *with $\epsilon_{abs} = .3$.*

From the exact eigenvalues, we determined the maximal absolute error in the eigenvalues as $\epsilon_{abs} = .3$. MATLAB's `poly` function computed the last coefficients with large absolute errors. In particular, $c_{20}^{poly}(U) \approx 10^6$ while the exact value is $c_{20}(U) = 1$. Figure 5.7 illustrates that for these last coefficients the quantities η_k^A from (5.2) do not bound the error from above. However, the "full order" error bounds do,

$$\eta_k^F \equiv \zeta_k + \gamma_{2n}\left(|c_k| + \zeta_k\right), \quad \text{where} \quad \zeta_k \equiv \sum_{i=1}^{k} \binom{n-k+i}{i} |c_{k-i}(U)|\, \epsilon_{abs}^i. \tag{5.6}$$

Acknowledgments. We thank the reviewers for helpful suggestions that improved the exposition of the paper.

REFERENCES

[1] F. B. Baker and M. R. Harwell, *Computing elementary symmetric functions and their derivatives: A didactic*, Appl. Psychol. Meas., 20 (1996), pp. 169–192.
[2] J. Demmel and P. Koev, *Accurate and efficient evaluation of Schur and Jack functions*, Math. Comp., 75 (2005), pp. 223–239.
[3] A. Eisinberg and G. Fedele, *A property of the elementary symmetric functions*, Calcolo, 42 (2005), pp. 31–36.
[4] G. H. Fischer, *Einführung in die Theorie Psychologischer Tests*, Huber, Bern, Switzerland, 1974.
[5] A. Galántai and C. J. Hegedüs, *Perturbation bounds for polynomials*, Numer. Math., 109 (2008), pp. 77–100.
[6] R. T. Gregory and D. L. Karney, *A Collection of Matrices for Testing Computational Algorithms*, Wiley Interscience, New York, 1969.
[7] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
[8] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.
[9] I. C. F. Ipsen and R. Rehman, *Perturbation bounds for determinants and characteristic polynomials*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 762–776.
[10] W. Li and W. Sun, *The perturbation bounds for eigenvalues of normal matrices*, Numer. Linear Algebra Appl., 12 (2005), pp. 89–94.
[11] D. S. Mitrinović, *Analytic Inequalities*, Springer, Heidelberg, Germany, 1970.
[12] R. Rehman, *Numerical Computation of the Characteristic Polynomial of a Complex Matrix*, Ph.D. thesis, North Carolina State University, Raleigh, NC, 2010.
[13] Y. Z. Song, *A note on the variation of the spectrum of an arbitrary matrix*, Linear Algebra Appl., 342 (2002), pp. 41–46.