Calcolo (2026) 63:8
https://doi.org/10.1007/s10092-026-00683-2

Perturbation theory and error analysis for the Cauchy formula

Pablo Díaz[1] · Yasmina Khiar[1] · Esmeralda Mainar[1] · Eduardo Royo-Amondarain[2]

Received: 19 August 2025 / Revised: 18 December 2025 / Accepted: 24 January 2026

Abstract

In this work, we analyze the numerical behavior of the classical Cauchy identity

$$\sum_{\lambda} s_{\lambda}(a_1,\ldots,a_n)s_{\lambda}(x_1,\ldots,x_m) = \prod_{j=1}^{n}\prod_{i=1}^{m}\frac{1}{1-a_jx_i},$$

by developing perturbation and running error analyses. We show that relative perturbations in the nodes x_i and coefficients a_j only induce small relative changes in the output provided some relative gaps are sufficiently large. We also propose an algorithm computing a posteriori relative error bound with low computational overhead. Finally, we derive truncation error bounds for the Schur expansion of the formula. Numerical experiments confirm the sharpness of the theoretical results and illustrate the effectiveness of the proposed bounds in practice.

Keywords Cauchy identity · Schur functions · Error analysis · Structured condition number

This work was partially supported by Spanish research grants PID2022-138569NB-I00 and RED2022-134176-T (MCI/AEI) and by Gobierno de Aragón (E41_23R).

✉ Eduardo Royo-Amondarain
eduroyo@unizar.es

Pablo Díaz
pablodiaz@unizar.es

Yasmina Khiar
ykhiar@unizar.es

Esmeralda Mainar
esmemain@unizar.es

1 Departamento de Matemática Aplicada/IUMA, Universidad de Zaragoza, Zaragoza, Spain

2 Departamento de Matemática Aplicada/CAPA, Universidad de Zaragoza, Zaragoza, Spain

Published online: 18 February 2026

1 Introduction

Recent advances in computational algebra and numerical linear algebra highlight two complementary challenges arising when symbolic systems are applied to large-scale or numerically sensitive problems. The first concerns the algorithmic complexity of computing Schur polynomials and related symmetric-function expansions efficiently [1–3]. The second pertains to numerical stability issues that arise when symmetric-function constructions are used to generate structured matrices, such as gramian or collocation matrices associated with symmetric polynomials [4, 5].

From a complexity-theoretic standpoint, the intrinsic difficulty of representing Schur polynomials by small algebraic formulas reveals fundamental limitations of determinant-based expressions. These limitations are not merely theoretical: in practice, they often manifest as severe numerical instabilities when such formulas are evaluated in floating-point arithmetic [1, 2]. In contrast, recent work exploiting bidiagonal factorizations and total-positivity properties derived from integrals of symmetric functions demonstrates that underlying structure can be leveraged to design algorithms achieving high relative accuracy in linear-algebra computations. Such approaches offer a promising framework for stable numerical evaluation and provide a natural bridge between symbolic constructions and reliable numerical algorithms.

Roundoff errors in N-digit binary floating-point arithmetic are studied using the standard floating-point model in Section 2.2 of [6]. In this model, the result of any arithmetic operation is assumed to differ from the exact result by a small relative error and loss of relative accuracy arises primarily when subtracting two nearly equal numbers of the same sign. This phenomenon, known as subtractive cancellation, is a major source of numerical instability in many linear algebra algorithms.

Cauchy-type identities are fundamental in the study of symmetric functions and algebraic combinatorics. They often involve expressions that link symmetric polynomials or functions, such as Schur functions, in a manner that reveals deep combinatorial and algebraic properties. Originally rooted in the classical works of Augustin-Louis Cauchy, these identities extend their relevance across various domains, including linear algebra, number theory, and special functions. Their versatility lies in their ability to establish connections between seemingly disparate entities, such as determinants, symmetric polynomials, and partition functions.

The classical Cauchy identity for Schur functions can be expressed as:

$$S := \sum_{\lambda} s_\lambda(a_1, \ldots, a_n) s_\lambda(x_1, \ldots, x_m) = \prod_{j=1}^{n} \prod_{i=1}^{m} \frac{1}{1 - a_j x_i}, \tag{1}$$

where the sum ranges over all partitions, and s_λ denote Schur functions. Notice that only the partitions λ with at most $\min(n, m)$ parts contribute to the sum since $s_\lambda(x_1, \ldots, x_n) = 0$ whenever the number of parts of λ exceeds the number of variables. This particular identity plays a fundamental role in the theory of symmetric functions linking generating functions to combinatorial structures.

The classical form as the expansion involving products of $(1 - x_i a_j)^{-1}$ is foundational. Moreover, the product

$$\prod_{j=1}^{n}\prod_{i=1}^{m}\frac{1}{1-a_jx_i} \tag{2}$$

is closely related to the determinant of the collocation matrix at the set of nodes $\{x_1,\ldots,x_n\}$ of the rational system $\left(\frac{1}{1-a_1x},\frac{1}{1-a_2x},\ldots,\frac{1}{1-a_nx}\right)$, for coefficients $a_i\in\mathbb{R}$, $i=1,\ldots,n$. In fact,

$$\det\left[\frac{1}{1-a_jx_i}\right]_{i,j=1}^{n}=\frac{\det V(x_1,\ldots,x_n)\,\det V(a_1,\ldots,a_n)}{\prod_{1\le i,j\le n}(1-a_jx_i)}, \tag{3}$$

where $V(x_1,\ldots,x_n)$ denotes the corresponding Vandermonde matrix.

In this paper, we perform a perturbation analysis of the product (2). It will be shown that the sensitivity of the product to small perturbations in the nodes and coefficients depends primarily on the relative separations of their products to 1, rather than on their absolute distances. This behavior is not unique to this product but also appears in the computation of other determinants. For instance, it is well known that the smallest relative gap,

$$\text{relgap}:=\min_{i\neq j}\frac{|x_i-x_j|}{|x_i|+|x_j|}, \tag{4}$$

provides a lower bound on the sensitivity of the determinant of $V(x_1,\ldots,x_n)$ (cf. [7]). Other interesting perturbation analyses can be found in [7–11].

Additionally, building on the performed structural sensitivity analysis, we develop a running error analysis for the computation of (2). This methodology not only allows the accurate evaluation of the Cauchy formula (1), but also provides realistic a posteriori error bounds that explicitly depend on the computed values. These error bounds are both computationally efficient and effective in tracking the propagation of numerical errors, making them particularly valuable in practical applications where the reliability and certification of computed results are essential.

The Schur polynomials play a crucial role in combinatorics, representation theory, mathematical physics, and multivariate statistics [12–15]. Several expressions for their computation can be found in [13, 15, 16]. Unfortunately, available formulas are either prohibitively inefficient, involving large sums of monomials, or deceptively simple but prone to severe numerical instability, such as determinant-based representations.

For a given partition $\lambda=(\lambda_1,\lambda_2,\ldots,\lambda_k)$ with $\lambda_1\ge\lambda_2\ge\cdots\ge\lambda_k\ge 0$ and $k\le n$, the classical definition of the Schur polynomial $s_\lambda(x_1,\ldots,x_n)$ as the ratio of determinants involving powers of variables:

$$s_\lambda(x_1,\ldots,x_n)=\frac{\det\left(x_i^{\lambda_j+n-j}\right)_{1\le i,j\le n}}{\det\left(x_i^{n-j}\right)_{1\le i,j\le n}}=\frac{\det\left(x_i^{\lambda_j+n-j}\right)_{1\le i,j\le n}}{\det V(x_1,\ldots,x_n)}, \tag{5}$$

is a very unstable way of computing it. Section 8 of [1] provides numerical examples that demonstrate its numerical instability. This instability is precisely what motivates the present work, where we avoid determinant-based evaluations and instead analyze the numerical behavior of the Cauchy product representation. In [3], interested readers can find insightful analyses of numerical instabilities arising in the context of symmetric polynomials within modern computational algebra systems; see, for instance, the discussion of determinant evaluations in SageMath's Symmetric Functions library.

The Schur polynomial $s_\lambda(x_1, \ldots, x_n)$ can also be written as

$$s_\lambda(x_1, \ldots, x_n) = \sum_T \prod_{i=1}^{n} x_i^{m_i(T)}, \tag{6}$$

where the sum is over all semistandard Young tableau T of shape λ with entries in $\{1, \ldots, n\}$, and $m_i(T)$ is the number of times i appears in T. This combinatorial definition of Schur polynomials (6) is always accurate since involves no subtractions (see [15]). To avoid the addition of $O\left(n^{|\lambda|}\right)$ terms, Algorithm 5.2 in [1], uses dynamic programming for the accurate and efficient computation in flotating-point arithmetic of $s_\lambda(x_1, \ldots, x_n)$, for $x_i > 0$. Its complexity is only linear in n and subexponential in $|\lambda|$. In fact, it is bounded by $O(e^{5.2\sqrt{|\lambda|}} \cdot \lambda_1 \cdot n)$ (see Proposition 5.3 of [1]).

If the left-hand side of the Cauchy identity is to be evaluated with a prescribed error tolerance, it is not necessary to sum over all partitions. Instead, a finite truncation of the sum may already achieve the desired level of accuracy. Taking into account this fact, we bound the error introduced by truncating the Schur function expansion.

The paper is organized as follows. Section 2 develops a perturbation analysis for the Cauchy formula, quantifying how small relative perturbations in the nodes and coefficients affect the value of the product (2), and deriving a structured condition number. In Sect. 3, a running error analysis is performed, providing an algorithm for computing the Cauchy formula along with a posteriori absolute and relative error bounds, using standard floating-point error models. Section 4 addresses the error introduced by truncating the Schur function expansion, and derives an upper bound for the tail of the sum based on combinatorial identities and asymptotic estimates. Finally, Sect. 5 provides numerical experiments validating the theoretical results. These include tests of the structured condition number, error tracking algorithms, and truncation bounds under different parameter regimes.

2 Perturbation theory

For a real value $x \in \mathbb{R}$, the value computed in a floating point arithmetic is usually denoted by either $\mathrm{fl}(x)$ or $\hat{x}$. Roundoff errors in N-digit binary floating-point arithmetic are studied using the standard floating point model introduced in Section 2.2 of [6]. In this model, the result of any arithmetic operation is assumed to differ from the exact result by a small relative error:

$$\mathrm{fl}(x \circ y) = (x \circ y)(1+\delta)^{\rho_i}, \qquad \circ \in \{+,-,\times,/\}, \quad |\delta| \le u, \tag{7}$$

where $\rho_i = \pm 1$ and $u := 2^{-N}$ is the machine precision. This model implies that products, quotients, and sums of like-signed quantities can be computed with low relative error. However, expressions involving cancellation may introduce significant error – for instance, the sum of three numbers cannot, in general, be computed accurately in this model [17].

Now, let us consider a set of nodes $\mathbf{x} = \{x_1, \ldots, x_m\}$ and a set of coefficients $\mathbf{a} = \{a_1, \ldots, a_n\}$ such that $0 < x_1 < \cdots < x_m \le 1$ and $0 < a_1 < \cdots < a_n \le 1$, with $a_n x_m \ne 1$.

The following perturbation analysis follows the same structural pattern as previous analyses in [7–11], in the sense that the sensitivity is controlled by relative gap conditions, and the rational structure of the factors $(1 - a_j x_i)^{-1}$ allows the perturbations to be accumulated multiplicatively using the standard floating-point model of Higham.

In order to obtain an appropriate structured condition number for the evaluation of (1), we establish a new relative distance:

$$\mathrm{relgap}_1 := \min \frac{|x_i - 1/a_j|}{|x_i|} = \frac{1}{a_n x_m} - 1, \tag{8}$$

and derive the following perturbation results.

Theorem 1 *Let S be defined as in (1) for nodes x_i, $i = 1, \ldots, m$, and coefficients a_j, $j = 1, \ldots, n$. Consider $\hat{S}$ as the value in (1) computed for perturbed nodes $\hat{x}_i = x_i(1+\delta_i)$, $i = 1, \ldots, m$, where the perturbations satisfy $|\delta_i| \le \delta$ with $0 < \delta < 1$. Then,*

$$\left| \frac{S - \hat{S}}{S} \right| \le \frac{\delta\, n\, m/\mu}{1 - \delta\, n\, m/\mu}, \tag{9}$$

for any $\mu \le \mathrm{relgap}_1$ as in (8).

Proof Observe that

$$\frac{(1 - a_j \hat{x}_i) - (1 - a_j x_i)}{1 - a_j x_i} = \frac{\hat{x}_i - x_i}{x_i - 1/a_j} = \frac{\delta_i x_i}{x_i - 1/a_j},$$

for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. Therefore, for any $\mu \le \mathrm{relgap}_1$, we can express

$$1 - a_j \hat{x}_i = (1 - a_j x_i)(1 + \delta_i'),$$

where $|\delta_i'| \le \delta/\mu$.

Using (1), and by accumulating the perturbations following Higham's standard technique (see Chapter 3 of [6]), we obtain

$$\hat{S} = S\prod_{j=1}^{n}\prod_{i=1}^{m}(1+\delta_i') = S(1+\bar{\delta}), \qquad |\bar{\delta}| \leq \frac{\delta\, nm/\mu}{1-\delta\, nm/\mu}, \tag{10}$$

and the result follows. □

Observe that, from (9), we derive

$$\left|\frac{S-\hat{S}}{S}\right| \leq n\, m\kappa\delta, \quad \kappa := \frac{1}{\mathrm{relgap}_1}, \tag{11}$$

and clearly see that small relative perturbations in the nodes only produce small relative changes in the value of *S* as long as the relative distance (8) of the product $a_n x_m$ to 1 is large enough.

The following result extends the previous analysis to the sum (1) computed with perturbed nodes and coefficients.

Theorem 2 *Let S be defined as in (1) for nodes x_i, $i = 1, \dots, m$, and coefficients a_j, $j = 1, \dots, n$. Consider $\hat{S}$ as the value in (1) computed for perturbed nodes $\hat{x}_i = x_i(1+\delta_i)$ and coefficients $\hat{a}_j = a_j(1+\delta_j')$, such that $|\delta_i|, |\delta_j'| \leq \delta$ with $0 < \delta < 1$, $i = 1, \dots, m$, $j = 1, \dots, n$. Then*

$$\left|\frac{S-\hat{S}}{S}\right| \leq \frac{2\delta n\, m/\mu}{1-2\delta n\, m/\mu} + \mathcal{O}(u^2), \tag{12}$$

for any $\mu \leq relgap_1$ as in (8).

Proof Since

$$\frac{(1-\hat{a}_j\hat{x}_i)-(1-a_jx_i)}{1-a_jx_i} = \frac{a_jx_i-\hat{a}_j\hat{x}_i}{1-a_jx_i} = \frac{x_i(\delta_j'+\delta_i+\delta_j'\delta_i)}{x_i-1/a_j},$$

for any $\mu \leq \mathrm{relgap}_1$, we can write

$$1-\hat{a}_j\hat{x}_i = (1-a_jx_i)(1+\delta_{i,j}'),$$

where $|\delta_{i,j}'| \leq (2\delta+\delta^2)/\mu = \delta(2+\delta)/\mu$. From (1), and accumulating the perturbations in the style of Higham (see Chapter 3 of [6]), we can write

$$\hat{S} = S\prod_{j=1}^{n}\prod_{i=1}^{m}(1+\delta_{i,j}') = S(1+\bar{\delta}), \quad |\bar{\delta}| \leq \frac{2\delta\, n\, m/\mu}{1-2\delta\, n\, m/\mu} + \mathcal{O}(u^2), \tag{13}$$

and the result follows. □

Now, from (12), the relative error satisfies

$$\left|\frac{S-\hat{S}}{S}\right| \leq 2\,n\,m\,\kappa\,\delta, \tag{14}$$

where κ is defined in (11). This inequality shows that small relative perturbations in the nodes and coefficients lead to small relative changes in the value of S, provided the relative gap between $a_n x_m$ and 1 given by (8) is sufficiently large. Thus, by (11), $n\,m\,\kappa$ can be interpreted as an appropriate structured condition number for computing (1) with respect to perturbations in the nodes; analogously, by (14), $2\,n\,m\,\kappa$ is an appropriate structured condition number if perturbations in both the nodes and the coefficients are considered.

Section 5 will illustrate the sensitivity of the sum to the proposed structured condition number.

3 Running error analysis

For the evaluation of the Cauchy formula (1), we propose the following algorithm.

Algorithm 1 Evaluation of the Cauchy formula (1)

```
Require: n, m, (a_1, ..., a_n), (x_1, ..., x_m)
  S_{0,1} ← 1
  for j = 1 : n do
      for i = 1 : m do
          S_{i,j} ← S_{i-1,j} / (1 - a_j * x_i)
      end for
      S_{0,j+1} ← S_{i,j}
  end for
```

Our goal is to adapt Algorithm 1 so that it simultaneously computes the determinant and provides a posteriori error bounds that account for the effects of finite-precision floating-point arithmetic. This enhancement is designed to incur only a negligible increase in computational cost. For a detailed exposition of the underlying methodology—whose notation we adopt hereafter—see [6]. For a practical application of a similar approach, we refer the reader to [18].

Let $E_{i,j}$ denote the error in the computation of $S_{i,j}$ using Algorithm 1, defined as

$$E_{i,j} := \tilde{S}_{i,j} - S_{i,j}, \quad i = 1, \ldots, m,\ j = 1, \ldots, n, \tag{15}$$

where $\tilde{S}_{i,j}$ is the computed value obtained through one multiplication ($*$), one subtraction ($-$), and one division ($/$).

According to the rounding model (7), the computed value $\tilde{S}_{i,j}$ can be expressed in terms of $\tilde{S}_{i-1,j}$ as

$$\tilde{S}_{i,j} = \tilde{S}_{i-1,j}\,(1 - a_j x_i(1+\delta_{i,j}))^{-1}(1+\bar{\delta}_{i,j})(1+u_{i,j})^{-1},$$

where $|\delta_{i,j}|, |\bar{\delta}_{i,j}|, |u_{i,j}| \leq u$, with u being the unit roundoff.

Using (15) and performing straightforward algebraic manipulations, we obtain

$$E_{i,j} + u_{i,j}\tilde{S}_{i,j} = E_{i-1,j}(1-a_jx_i)^{-1} + \tilde{S}_{i-1,j}(1-a_jx_i)^{-1}\left[\delta_{i,j}\frac{a_jx_i}{1-a_jx_i} + \bar{\delta}_{i,j} + \mathcal{O}(u^2)\right].$$

Consequently, the error $|E_{i,j}|$ satisfies the bound

$$|E_{i,j}| \leq |E_{i-1,j}|\,|1-a_jx_i|^{-1} + u\left[|\tilde{S}_{i-1,j}|\,|1-a_jx_i|^{-1}\left(1+\frac{|a_jx_i|}{|1-a_jx_i|}\right) + |\tilde{S}_{i,j}|\right] + \mathcal{O}(u^2).$$

Since $S_{0,1}$ is computed exactly (no arithmetic operations are involved), we have $E_{0,1} = 0$, and therefore

$$|E_{i,j}| \leq u\,\pi_{i,j} + \mathcal{O}(u^2), \tag{16}$$

where the majorizing sequence $\pi_{i,j}$ is defined recursively by

$$\pi_{0,1} := 0, \quad \pi_{i,j} := \left(\pi_{i-1,j} + |\tilde{S}_{i-1,j}|\right)|1-a_jx_i|^{-1} + |\tilde{S}_{i-1,j}|\,|1-a_jx_i|^{-2}|a_jx_i| + |\tilde{S}_{i,j}|, \tag{17}$$

for $j = 1, \ldots, n$ and $i = 1, \ldots, m$.

To slightly reduce the computational cost of evaluating (17), we introduce the auxiliary sequence

$$M_{i,j} := \frac{\pi_{i,j} + |\tilde{S}_{i,j}|}{2}, \quad j = 1, \ldots, n, \; i = 1, \ldots, m,$$

which satisfies the recursion

$$M_{0,1} := \frac{|\tilde{S}_{0,1}|}{2}, \quad M_{i,j} = M_{i-1,j}|1-a_jx_i|^{-1} + |\tilde{S}_{i-1,j}|\,|1-a_jx_i|^{-2}\frac{|a_jx_i|}{2} + |\tilde{S}_{i,j}|,$$

for $j = 1, \ldots, n$ and $i = 2, \ldots, m$.

Finally, from (16) we derive the running absolute error bound

$$|E_{m,n}| = |\tilde{S}_{m,n} - S_{m,n}| \leq \gamma\,u + \mathcal{O}(u^2), \quad \gamma := 2M_{m,n} - |\tilde{S}_{m,n}|,$$

where $\gamma\,u$ is computed alongside the determinant evaluation in Algorithm 1.

Using the following lemma, we can derive a sufficient condition–based on the computed product $\tilde{S}_{n,m}$ and the absolute error bound $\gamma\,u$–to obtain a running relative error bound. The proof is omitted, as it follows by an argument, which is analogous to that of Theorem 3.1 in [19].

Lemma 1 *Let $\tilde{x}$ denote the floating-point approximation of a nonzero real number x, and assume that*

$$|\tilde{x} - x| \leq uK + \mathcal{O}(u^2),$$

where u is the unit roundoff. If $|\tilde{x}| > uK$, then

$$\left|\frac{\tilde{x} - x}{x}\right| \leq u\frac{K}{|\tilde{x}|} + \mathcal{O}(u^2). \tag{18}$$

Let us observe that, as a direct consequence of Lemma 1, if $|\tilde{S}_{n,m}| > u\big(2M_{n,m} - |\tilde{S}_{n,m}|\big)$, then the quantity

$$\Gamma := \frac{u\big(2M_{n,m} - |\tilde{S}_{n,m}|\big)}{|\tilde{S}_{n,m}|} \tag{19}$$

provides a valid running relative error bound for the computation of S. This bound can be computed alongside the determinant evaluation by using Algorithm 2, which is a suitably modified version of Algorithm 1.

Algorithm 2 Evaluation of the Cauchy formula (1) with running error bound

---

Require: $n, m, (a_1, \ldots, a_n), (x_1, \ldots, x_m)$
  $S_{0,1} \leftarrow 1$
  $M_{0,1} \leftarrow 1/2$
  for $j = 1 : n$ **do**
    for $i = 1 : m$ **do**
      $S_{i,j} \leftarrow S_{i-1,j}/|1 - a_j * x_i|$
      $M_{i,j} \leftarrow M_{i-1,j}/|1 - a_j * x_i| + |S_{i,j-1}| * |a_j * x_i|/[2 * (1 - a_j * x_i)^2] + |S_{i,j}|$
    end for
    $S_{0,j+1} \leftarrow S_{i,j}$
  end for
  $\gamma \leftarrow u(2M_{n,m} - |S_{n,m}|)$
  $\Gamma \leftarrow \gamma/|S_{n,m}|$

Section 5 illustrates the accuracy and sharpness of the running relative error bound given in (19).

4 Upper error bound of finite truncations

If the left-hand side of the Cauchy identity

$$\sum_{\lambda} s_\lambda(a_1, \ldots, a_n) s_\lambda(x_1, \ldots, x_m) = \prod_{j=1}^{n} \prod_{i=1}^{m} \frac{1}{1 - a_j x_i},$$

can be evaluated up to an error tolerance $\Delta \geq 0$, it is unnecessary to sum over all partitions, and a finite truncation of the sum may already yield sufficient accuracy.

For a positive integer K, we can split the sum as

$$\sum_{\lambda\in\Lambda} s_\lambda(\mathbf{a}_n)s_\lambda(\mathbf{x}_m) = S_K(\mathbf{a}_n,\mathbf{x}_m) + R_K(\mathbf{a}_n,\mathbf{x}_m), \tag{20}$$

where we have used the shorthand notation $\mathbf{a}_n := a_1,\ldots,a_n$ and $\mathbf{x}_m := x_1,\ldots,x_m$, and

$$S_K(\mathbf{a}_n,\mathbf{x}_m) := \sum_{k=0}^{K}\sum_{|\lambda|=k} s_\lambda(\mathbf{a}_n)s_\lambda(\mathbf{x}_m), \quad R_K(\mathbf{a}_n,\mathbf{x}_m) := \sum_{k=K+1}^{\infty}\sum_{|\lambda|=k} s_\lambda(\mathbf{a}_n)s_\lambda(\mathbf{x}_m).$$

In this section, we derive a bound for the tail of the sum, $R_K(\mathbf{a}_n,\mathbf{x}_m)$, proving that $\lim_{K\to\infty} R_K(\mathbf{a}_n,\mathbf{x}_m) = 0$; this immediately allows to determine a value of K that guarantees $R_K(\mathbf{a}_n,\mathbf{x}_m) \le \Delta$.

For this purpose, we shall use the following identity.

Lemma 2 *For* $k,n,m\in\mathbb{N}$,

$$\sum_{|\lambda|=k} s_\lambda(\underbrace{1,\ldots,1}_{n})s_\lambda(\underbrace{1,\ldots,1}_{m}) = \binom{k+nm-1}{k}. \tag{21}$$

Proof Let us consider tuples $(x_1,\ldots,x_n) = (t,\ldots,t)$ and $(y_1,\ldots,y_m) = (1,\ldots,1)$. From the well-known homogeneity property of Schur polynomials, described by

$$s_\lambda(tx_1,\ldots,tx_l) = t^{|\lambda|}s_\lambda(x_1,\ldots,x_l), \quad t\in\mathbb{R}, \tag{22}$$

the corresponding Cauchy identity (1) reads as follows

$$\sum_{k=0}^{\infty} t^k \sum_{|\lambda|=k} s_\lambda(\underbrace{1,\ldots,1}_{n})s_\lambda(\underbrace{1,\ldots,1}_{m}) = \left(\frac{1}{1-t}\right)^{nm}. \tag{23}$$

Recall that the complete homogeneous symmetric polynomial of degree k in l variables, usually denoted as $h_k(x_1,\ldots,x_l)$ for $k=0,1,2,\ldots$, is the sum of all monomials of total degree k in the variables. Formally,

$$h_k(x_1,\ldots,x_l) := \sum_{1\le i_1\le i_2\le\cdots\le i_k\le l} x_{i_1}x_{i_2}\cdots x_{i_k}.$$

Taking into account that the generating function for $h_k(x_1,\ldots,x_l)$ is given by the infinite product

$$\sum_{k=0}^{\infty} h_k(x_1,\ldots,x_l)t^k = \prod_{i=1}^{l}\frac{1}{1-x_i t} \tag{24}$$

(see [13, Chapter 1.4]), and considering $l = nm$ and $x_i = 1$, $i = 1, \ldots, nm$, we derive

$$\sum_{k=0}^{\infty} h_k(\underbrace{1, \ldots, 1}_{nm})t^k = \left(\frac{1}{1-t}\right)^{nm} = \sum_{k=0}^{\infty} t^k \sum_{|\lambda|=k} s_\lambda(\underbrace{1, \ldots, 1}_{n}) s_\lambda(\underbrace{1, \ldots, 1}_{m}). \quad (25)$$

Equating the coefficient of any t^k in the previous identity, we derive

$$\sum_{|\lambda|=k} s_\lambda(\underbrace{1, \ldots, 1}_{n}) s_\lambda(\underbrace{1, \ldots, 1}_{m}) = h_k(\underbrace{1, \ldots, 1}_{nm}) = \binom{k+nm-1}{k}, \quad (26)$$

where the last equality in (26) follows from the fact that evaluating at 1 the complete symmetric function gives the number of monomials of total degree k in nm variables, that is, the number of combinations with repetition of choosing k elements from nm distinct ones. □

Now, we prove the main result of this section.

Theorem 3 *For $K > 0$ and any sequences $0 < a_1 < \cdots < a_n < 1$ and $0 < x_1 < \cdots < x_m < 1$, we have*

$$\begin{aligned} &\sum_{k=K+1}^{\infty} \sum_{|\lambda|=k} s_\lambda(a_1, \ldots, a_n) s_\lambda(x_1, \ldots, x_m) \\ &< (a_n x_m)^K \sum_{k=0}^{nm-1} \binom{K+nm}{k} \left(\frac{a_n x_m}{1 - a_n x_m}\right)^{nm-k}. \end{aligned} \quad (27)$$

Proof Let us denote $\rho := a_n x_m$ and $N := nm$. Using the combinatorial definition of Schur polynomials (6) and their homogeneity property (22), we derive

$$\begin{aligned} &\sum_{k=K+1}^{\infty} \sum_{|\lambda|=k} s_\lambda(a_1, \ldots, a_n) s_\lambda(x_1, \ldots, x_m) \\ &< \sum_{k=K+1}^{\infty} \sum_{|\lambda|=k} \rho^k s_\lambda(\underbrace{1, \ldots, 1}_{n}) s_\lambda(\underbrace{1, \ldots, 1}_{m}). \end{aligned} \quad (28)$$

From Lemma 2, the sum over partitions of size k can be evaluated exactly. Then,

$$\sum_{k=K+1}^{\infty} \sum_{|\lambda|=k} s_\lambda(a_1, \ldots, a_n) s_\lambda(x_1, \ldots, x_m) < \sum_{k=K+1}^{\infty} \rho^k \binom{k+N-1}{k}. \quad (29)$$

Note that the sum at the right-hand side of (29) can be expressed in terms of derivatives of a geometric series. In fact,

$$\sum_{k=K+1}^{\infty} \binom{k+N-1}{k} \rho^k = \frac{1}{(N-1)!} \sum_{k=K+1}^{\infty} (k+N-1)\cdots(k+1)\rho^k$$
$$= \frac{1}{(N-1)!} \frac{d^{N-1}}{d\rho^{N-1}} \left(\sum_{k=K+1}^{\infty} \rho^{k+N-1} \right) = \frac{1}{(N-1)!} \frac{d^{N-1}}{d\rho^{N-1}} \left(\frac{\rho^{N+K}}{1-\rho} \right). \tag{30}$$

Finally, applying the Leibniz rule for higher-order derivatives yields the stated result. □

The behavior of the explicit upper bound given by expression (27) becomes clearer when viewed asymptotically. The function

$$f(\rho; K, N) = \rho^K \sum_{k=0}^{N-1} \binom{K+N}{k} \left(\frac{\rho}{1-\rho} \right)^{N-k}, \quad \rho \in (0,1), \tag{31}$$

approaches 0 as $K \to \infty$. Moreover, it can be checked that

$$f(\rho; K', N) < f(\rho; K, N), \quad K' > K.$$

So, for any $\varepsilon > 0$, and any given ρ and N, there always exists $K > 0$ so that for $K' \geq K$ we have $f(\rho; K', N) < \varepsilon$.

Let $\rho = a_n x_m < 1$ and $N := nm$. From (27), for fixed N and ρ, the truncation error decays essentially geometrically with rate ρ^K when $K \to \infty$. In particular, when ρ is moderately small, the remainder decreases extremely rapidly, which is consistent with the behavior observed in the numerical experiments of Sect. 5. This interpretation also explains why the value of K required to achieve a prescribed tolerance is primarily governed by the magnitude of the extreme product $a_n x_m$.

In the following section, the convergence of the sum over partitions together with the behavior of bound (27) are numerically explored.

5 Numerical experiments

In this section we present a series of numerical experiments that analyze several aspects of the previously discussed theoretical results, and that are useful for exploring their strengths and limitations. Regarding the methodology of the presented results, for a given real quantity x, the relative error of its computed value $\hat{x}$ is given by $|\hat{x} - x|/|x|$. The exact value x is determined by means Wolfram Mathematica, using 100-digit precision arithmetics.

First, we study the accuracy of the right-hand side of formula (1) for increasing values of the structured condition number deduced in Sect. 2. Moreover, we pursue to test the accuracy of Algorithm 1 under numerically challenging conditions, as well as the usefulness of the relative running error bound provided by Algorithm 2. To achieve these goals, strictly increasing equidistant nodes $x_i = i/(m+\epsilon)$, $i = 1, \ldots, m$ and coefficients $a_j = j/(n+\epsilon)$, $j = 1, \ldots, n$ are considered. Notice that, for small values of ϵ, $a_n x_m$ approaches unity as *n*, *m* grow, and therefore a lack of accuracy in the computation of the product due to cancellation is expected—indeed, this behavior can be observed in Table 1. In addition, the relative running error bound computed by means of Algorithm 2 exhibits a sharp performance, endorsing its use in keeping track of the accuracy loss.

Secondly, with regard to the left-hand side of formula (1), our interest is to study the convergence speed when considering the sum over partitions of increasing size $|\lambda|$. To do so, we will consider the a priori adverse scenario of relatively close nodes and coefficients, and for which the bound provided by Theorem 3 is expected to be not far of the actual error. Two experiments are provided, analyzing the dependence of the results with both the size $n \times m$ and the value $a_n x_m$.

In Table 2, nodes $x_i = 1/10 + (i-m)/1000$, $i = 1, \ldots, m$ and coefficients $a_j = 3/10 + (j-n)/1000$, $j = 1, \ldots, n$ are considered, keeping $a_n x_m$ size-independent. The contribution to the sum of the left-hand side of (1) provided by

$$\sum_{|\lambda|=k} s_\lambda(\mathbf{a}_n) s_\lambda(\mathbf{x}_m),$$

corresponding to the partitions of size $|\lambda| = k$, is gathered, together with the relative error of $S_k(\mathbf{a}_n, \mathbf{x}_m)$—the truncated sum as defined in (20). Finally, the relative error bound provided by (27) is also depicted. As can be seen, the contributions to the sum are not necessarily decreasing with *k*; in fact, they seem to have a maximum highlighted in bold in the table which depends on the size $n \times m$: as the dimension of the system grows, so does the size of the partitions that contribute the most. In addition, the provided bound also captures this behavior.

In the next experiment, to study the dependence with $a_n x_m$, nodes $x_i = \alpha/10 + (i-m)/1000$, $i = 1, \ldots, m$, $\alpha = 2, 3, 5$, and coefficients $a_j = 2/10 + (j-n)/1000$, $j = 1, \ldots, n$ are considered, keeping a constant size $n \times m = 8 \times 12$. The same quantities as in the previous case are collected in Table 3, where a similar behavior is observed: the dependence of the contributions of fixed partition size $|\lambda| = k$ exhibits a maximum on *k*, which position grows with $a_n x_m$.

Table 1 Structured condition number, relative error, and relative error bound in the computation of S through formula (1) with $x_i = i/(m+\epsilon), i = 1, \ldots, m$ and $a_j = j/(n+\epsilon)$, $j = 1, \ldots, n$

$n \times m$	ϵ	$nm\kappa$	Formula (1)	Running error
20×10	1	$1.3e+03$	$3.8e-15$	$6.7e-14$
20×10	10^{-1}	$1.3e+04$	$2.5e-14$	$9.2e-14$
20×10	10^{-3}	$1.3e+06$	$7.0e-13$	$1.6e-12$
20×10	10^{-6}	$1.3e+09$	$3.1e-10$	$1.5e-09$
40×20	1	$1.0e+04$	$1.3e-14$	$2.8e-13$
40×20	10^{-1}	$1.1e+05$	$1.0e-13$	$3.3e-13$
40×20	10^{-3}	$1.1e+07$	$1.2e-12$	$3.3e-12$
40×20	10^{-6}	$1.1e+10$	$4.5e-10$	$3.0e-09$

Table 2 Dependence on $n \times m$ of the contributions to the sum over partitions of fixed size *k*, the relative error of the sum up to size *k* and the relative error bound given by (27), for $x_i = 1/10 + (i - m)/1000$, $a_j = 3/10 + (j - n)/1000$, $i = 1, \ldots, m$, $j = 1, \ldots, n$, with $a_n x_m = 0.03$. For each size $\boldsymbol{n} \times \boldsymbol{m}$, the row that contributes the most to the sum over partitions of fixed ***k*** is highlighted in bold

$n \times m$	k	$\sum_{\lvert\lambda\rvert=k} s_\lambda(\mathbf{a}_n)s_\lambda(\mathbf{x}_m)$	Relative error of $S_k(\mathbf{a}_n, \mathbf{x}_m)$	Relative bound (27)
5×5	**1**	$\mathbf{7.3e-01}$	$\mathbf{1.8e-01}$	$\mathbf{5.4e-01}$
	2	$2.8e-01$	$4.3e-02$	$1.9e-01$
	3	$7.3e-02$	$8.5e-03$	$4.7e-02$
	4	$1.5e-02$	$1.4e-03$	$9.5e-03$
	5	$2.5e-03$	$2.0e-04$	$1.6e-03$
	10	$5.9e-08$	$2.9e-09$	$4.1e-08$
	15	$2.4e-13$	$9.0e-15$	$1.9e-13$
8×12	1	$2.7e+00$	$7.6e-01$	$1.2e+00$
	2	$\mathbf{3.7e+00}$	$\mathbf{5.2e-01}$	$\mathbf{9.6e-01}$
	3	$3.3e+00$	$3.0e-01$	$6.9e-01$
	4	$2.3e+00$	$1.5e-01$	$4.2e-01$
	5	$1.3e+00$	$6.5e-02$	$2.2e-01$
	10	$8.6e-03$	$2.0e-04$	$1.5e-03$
	15	$6.1e-06$	$9.4e-08$	$1.4e-06$
12×19	1	$6.1e+00$	$9.9e-01$	$2.1e+00$
	2	$1.9e+01$	$9.5e-01$	$2.1e+00$
	3	$3.9e+01$	$8.7e-01$	$2.1e+00$
	4	$6.0e+01$	$7.5e-01$	$1.9e+00$
	5	$7.4e+01$	$6.0e-01$	$1.8e+00$
	6	$\mathbf{7.7e+01}$	$\mathbf{4.4e-01}$	$\mathbf{1.5e+00}$
	7	$6.9e+01$	$3.0e-01$	$1.2e+00$
	8	$5.4e+01$	$1.9e-01$	$8.7e-01$
	9	$3.8e+01$	$1.1e-01$	$5.9e-01$
	10	$2.4e+01$	$5.8e-02$	$3.8e-01$
	15	$7.4e-01$	$9.9e-04$	$1.5e-02$

6 Conclusions and further research

In this work, we have developed a comprehensive perturbation and error analysis for the classical Cauchy identity in terms of Schur functions, deriving structured condition numbers that accurately quantify its sensitivity with respect to perturbations in both the nodes and the coefficients. Our results show that the relative error during the floating-point computation is governed by the relative gap between the extreme product $a_n x_m$ and 1, and that small perturbations induce only moderate relative changes in the output when this gap is sufficiently large.

An efficient algorithm with a low-cost running a posteriori relative error bound is also proposed. The provided bound, computed alongside the main evaluation, has negligible additional computational cost and reliably tracks the propagation of rounding errors in floating-point arithmetic. Numerical experiments confirm its sharpness and practical usefulness.

We have also addressed the problem of truncating the Schur expansion of the Cauchy identity, providing explicit upper bounds for the truncation error. The numerical

Table 3 Dependence on $a_n x_m$ of the contributions to the sum over partitions of fixed size k, the relative error of the sum up to size k and the relative error bound given by (27), for $x_i = \alpha/10 + (i-m)/1000$, $i = 1, \ldots, m$, $\alpha = 2, 3, 5$, and $a_j = 2/10 + (j-n)/1000$, $j = 1, \ldots, n$, for $n \times m = 8 \times 12$. For each size $\boldsymbol{n} \times \boldsymbol{m}$, the row that contributes the most to the sum over partitions of fixed $\boldsymbol{k}$ is highlighted in bold

$a_n x_m$	k	$\sum_{\lvert\lambda\rvert=k} s_\lambda(\mathbf{a}_n)s_\lambda(\mathbf{x}_m)$	Relative error of $S_k(\mathbf{a}_n, \mathbf{x}_m)$	Relative bound (27)
0.04	1	$3.7e+00$	$8.9e-01$	$1.2e+00$
	2	$6.8e+00$	$7.3e-01$	$1.1e+00$
	3	$\mathbf{8.5e+00}$	$\mathbf{5.3e-01}$	$\mathbf{9.0e-01}$
	4	$8.0e+00$	$3.4e-01$	$6.7e-01$
	5	$6.1e+00$	$1.9e-01$	$4.4e-01$
	10	$1.9e-01$	$2.5e-03$	$1.1e-02$
	15	$6.4e-04$	$5.3e-06$	$4.1e-05$
0.06	1	$5.6e+00$	$9.8e-01$	$1.2e+00$
	2	$1.6e+01$	$9.3e-01$	$1.2e+00$
	3	$2.9e+01$	$8.3e-01$	$1.2e+00$
	4	$4.2e+01$	$6.9e-01$	$1.1e+00$
	5	$\mathbf{4.9e+01}$	$\mathbf{5.3e-01}$	$\mathbf{9.0e-01}$
	6	$4.8e+01$	$3.8e-01$	$7.1e-01$
	7	$4.0e+01$	$2.5e-01$	$5.1e-01$
	8	$3.0e+01$	$1.5e-01$	$3.4e-01$
	9	$2.0e+01$	$8.3e-02$	$2.1e-01$
	10	$1.2e+01$	$4.3e-02$	$1.2e-01$
	15	$3.2e-01$	$6.7e-04$	$3.0e-03$
0.10	1	$9.3e+00$	$1.0e+00$	$1.4e+00$
	2	$4.4e+01$	$1.0e+00$	$1.4e+00$
	3	$1.4e+02$	$9.9e-01$	$1.3e+00$
	4	$3.4e+02$	$9.7e-01$	$1.3e+00$
	5	$6.5e+02$	$9.4e-01$	$1.3e+00$
	6	$1.1e+03$	$8.8e-01$	$1.3e+00$
	7	$1.5e+03$	$7.9e-01$	$1.2e+00$
	8	$1.9e+03$	$6.9e-01$	$1.1e+00$
	9	$2.1e+03$	$5.7e-01$	$9.8e-01$
	10	$\mathbf{2.2e+03}$	$\mathbf{4.6e-01}$	$\mathbf{8.3e-01}$
	11	$2.0e+03$	$3.5e-01$	$6.7e-01$
	12	$1.8e+03$	$2.5e-01$	$5.2e-01$
	13	$1.4e+03$	$1.7e-01$	$3.8e-01$
	14	$1.1e+03$	$1.1e-01$	$2.7e-01$
	15	$7.6e+02$	$7.1e-02$	$1.8e-01$

tests illustrate both the asymptotic decay of the remainder and the dependence of the main contributions on the partition size.

Several directions for future work remain open. The methodology developed here could be extended to other Cauchy-type identities involving different families of symmetric functions (e.g., Hall–Littlewood, Macdonald, or Jack polynomials), where stability and truncation issues may be even more delicate. The integration of our error bounds into certified numerical libraries for symmetric functions could make these results directly applicable in computational algebra systems and scientific computing environments.

Acknowledgements The authors gratefully acknowledge Gregory Olshansky for helpful discussions, and an anonymous referee for valuable suggestions.

Author contributions The authors contributed equally to this work.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. Partial financial support was received from the Spanish research grants PID2022-138569NB-I00 and RED2022-134176-T (MCI/AEI) and by Gobierno de Aragón (E41_23R).

Data availability The source code used to run the numerical experiments is available upon request.

Declarations

Conflict of interest The authors have no conflict of interest to declare that are relevant to the content of this article.

References

1. Demmel, J., Koev, P.: Accurate and efficient evaluation of Schur and Jack functions. Math. Comput. **75**(253), 223–239 (2006). https://doi.org/10.1090/S0025-5718-05-01780-1
2. Chaugule, P., Kumar, M., Limaye, N., Mohapatra, C.K., She, A., Srinivasan, S.: Schur polynomials do not have small formulas if the determinant does not. Comput. Complex. (2023). https://doi.org/10.1007/s00037-023-00236-x
3. SageMath Developers: Symmetric Functions in SageMath. Online documentation (2025). https://doc.sagemath.org/html/en/reference/combinat/sage/combinat/sf/sf.html
4. Díaz, P., Mainar, E., Rubio, B.: Polynomial total positivity and high relative accuracy through Schur polynomials. J. Sci. Comput. **97**(1), 10 (2023). https://doi.org/10.1007/s10915-023-02323-1
5. Díaz, P., Mainar, E., Rubio, B.: Total positivity, Gramian matrices, and Schur polynomials. AIMS Math. **10**(2), 2375–2391 (2025). https://doi.org/10.3934/math.2025110
6. Higham, N.J.: Accuracy and Stability of Numerical Algorithms, 2nd edn. Society for Industrial and Applied Mathematics, SIAM, Philadelphia (2002). https://doi.org/10.1137/1.9780898718027
7. Demmel, J., Koev, P.: Accurate SVDs of polynomial Vandermonde matrices involving orthonormal polynomials. Linear Algebra Appl. **417**, 382–396 (2006). https://doi.org/10.1016/j.laa.2005.09.014
8. Koev, P.: Accurate Eigenvalues and SVDs of totally nonnegative matrices. SIAM J. Matrix Anal. Appl. **27**(1), 1–23 (2005). https://doi.org/10.1137/S0895479803438225
9. Marco, A., Martínez, J.-J.: Accurate computations with totally positive Bernstein-Vandermonde matrices. Electron. J. Linear Algebra **26**, 357–380 (2013). https://doi.org/10.13001/1081-3810.1658
10. Marco, A., Martínez, J.-J., Viaña, R.: Accurate polynomial interpolation by using the Bernstein basis. Numer. Algor. **75**(3), 655–674 (2017). https://doi.org/10.1007/s11075-016-0215-7

11. Marco, A., Martínez, J.-J.: Accurate computations with Said-Ball-Vandermonde matrices. Linear Algebra Appl. **432**(11), 2894–2908 (2010). https://doi.org/10.1016/j.laa.2009.12.034
12. James, G., Kerber, A.: The Representation Theory of the Symmetric Group. Addison-Wesley Publishing Co., Reading (1981)
13. Macdonald, I.G.: Symmetric Functions and Hall Polynomials, 2nd edn. Oxford University Press, New York (1998)
14. Muirhead, R.J.: Aspects of Multivariate Statistical. Theory Wiley Series in Probability and Mathematical Statistics. Wiley, New York (1982)
15. Stanley, R.P., Fomin, S.: Enumerative Combinatorics. Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge (1999)
16. Macdonald, I.G.: Schur functions: theme and variations. Séminaire Lotharingien de Combinatoire [electronic only] **28**, 5–39 (1992)
17. Demmel, J., Koev, P.: Necessary and sufficient conditions for accurate and efficient rational function evaluation and factorizations of rational matrices. In: Structured Matrices in Mathematics, Computer Science, and Engineering, II (Boulder, CO, 1999). Contemp. Math., vol. 281, pp. 117–143. Amer. Math. Soc., Providence, RI (2001). https://doi.org/10.1090/conm/281/04652
18. Mainar, E., Peña, J.M.: Corner cutting algorithms associated with optimal shape preserving representations. Comput. Aided Geom. Des. **16**(9), 883–906 (1999). https://doi.org/10.1016/S0167-8396(99)00035-7
19. Delgado, J., Peña, J.M.: Running relative error for the evaluation of polynomials. SIAM J. Sci. Comput. **31**(5), 3905–3921 (2009). https://doi.org/10.1137/080745249