Linear Algebra and its Applications 745 (2026) 54–85

Contents lists available at ScienceDirect

Linear Algebra and its Applications

journal homepage: www.elsevier.com/locate/laa

Parameterizing intersecting surfaces via invariants

Timon S. Gutleb [a,*], Rhyan Barrett [b], Julia Westermayr [b,c], Christoph Ortner [d]

[a] *School of Computer Science, University of Leeds, Leeds, LS2 9JT, UK*
[b] *Wilhelm-Ostwald-Institut für Physikalische und Theoretische Chemie, Universität Leipzig, Germany*
[c] *ScaDS.AI (Center for Scalable Data Analytics and Artificial Intelligence) Dresden/Leipzig, Humboldtstraße 25, 04105 Leipzig, Germany*
[d] *Department of Mathematics, University of British Columbia, Vancouver, V6T1Z2, BC, Canada*

ARTICLE INFO

Article history:
Received 4 July 2024
Received in revised form 14 April 2026
Accepted 11 May 2026
Available online 13 May 2026
Submitted by V. Mehrmann

MSC:
65H04
65H05

Keywords:
Companion matrix
Surface crossings
Orthogonal polynomials
Excited states

ABSTRACT

We introduce and analyze numerical companion matrix methods for the reconstruction of hypersurfaces with crossings from smooth interpolants given unordered or, without loss of generality, value-sorted data. The problem is motivated by the desire to machine learn potential energy surfaces arising in molecular excited state computational chemistry applications. We present simplified models which reproduce the analytically predicted convergence and stability behaviors as well as two application-oriented numerical experiments: the electronic excited states of Graphene featuring Dirac conical cusps and energy surfaces corresponding to a sulfur dioxide (SO_2) molecule in different configurations.

* Corresponding author.
E-mail address: t.s.gutleb@leeds.ac.uk (T.S. Gutleb).

https://doi.org/10.1016/j.laa.2026.05.009

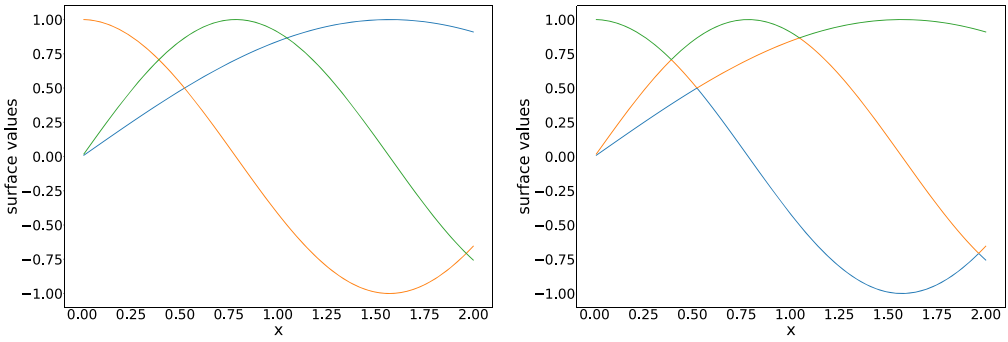


Fig. 1. Intersecting, smooth sinusoidal curves in (a), sorted by value in (b), revealing non-smooth cusps in the "j-th entry by value" function.

1. Introduction

Branching spaces, otherwise known as *conical intersections*, are crucial for determining the dynamics of molecules and materials upon light excitation, molecular orbitals and material band structure. Their accurate description is especially important to advance our understanding of fundamental reactions like photosynthesis or vision [45]. The relevant regions, however, are characterized by avoided crossings, i.e., two hypersurfaces that touch and become degenerate, but do not cross. This leads to non-smooth hyper-surfaces, making common machine learning models fail in their accurate modeling [77].

The aim of this paper is thus to describe and analyze numerical methods for reconstructing multiple intersecting real-valued hyper-surfaces from unordered data using globally smooth interpolations. By *unordered* we mean that the information "which surface" an evaluation point belongs to is unavailable. All surfaces throughout this work will be real-valued graphs. Our focus will be on theoretical aspects in a relatively idealized setting.

By means of example, Fig. 1(a) shows three real analytic curves, which are then value-sorted in (b) resulting in cusps at intersection points. The latter is, e.g., the form of the adiabatic potential energy surfaces produced in computational quantum chemistry. Well-known results on polynomial approximation, which we review in Sections 3 and 4, tell us that the best-approximation rate in this case will be $O(1/n)$, where n is the polynomial degree. On the other hand, if we knew the original analytic curves, i.e., if we could "disentangle" the intersections, then standard polynomial approximation schemes would guarantee an exponential or even super-exponential rate.

Our investigation of reconstructing multiple intersecting graphs is motivated by applications in computational chemistry, where touching adiabatic potential energy surfaces of molecules are frequently studied using machine learning algorithms [3,12,18,37,77,78] relying on smooth interpolants but analogous challenges also arise for interpolating band diagrams in materials science applications [54,58,85]. In most of these applications the

point-wise surface values are given as a list of real eigenvalues of a self-adjoint operator, with degenerate eigenvalues corresponding to the hypersurface crossings we are concerned with. The spectrum of this operator is typically expensive to compute, which motivates the need to construct computationally efficient surrogates.

To circumvent the challenge posed by cusps near crossings of value-sorted surfaces, we introduce an approximation scheme based on approximating smooth invariants. For example, suppose the graphs are given by $f_i(\mathbf{x}), i = 1, \dots, m$, $\mathbf{x} \in \mathbb{R}^d$ then the power sum polynomials

$$p_k(\mathbf{x}) = \sum_{j=1}^{m} f_j(\mathbf{x})^k, \qquad k = 1, \dots, m$$

are smooth functions of $\mathbf{x}$ (at least in the setting of Fig. 1). Moreover, the power sum polynomials form a complete set of invariants: from the values $(p_k)_{k=1}^m$ one can, in principle, reconstruct the surface values $(f_i)_{i=1}^m$. By approximating the smooth p_k we can recover a fast convergence rate under general and natural assumptions on the graphs f_i. Given an unseen input $\mathbf{x}$ one can then first predict the invariants $p_k(\mathbf{x})$ from which approximations to the surfaces $f_i(\mathbf{x})$ can then be reconstructed.

In practice, this procedure is numerically subtle, and the purpose of this work is therefore to explore the numerical challenges that arise in its implementation. For example, we will work with a different set of invariants (the elementary symmetric polynomials) for which the reconstruction step can be formulated in terms of a companion matrix eigenvalue problem equivalent to root-finding for certain polynomials. In this setting there are several polynomial bases and companion matrix representations to choose from which are natural for different reasons and have distinct behavior in numerical practice.

Machine learning seams of conical intersection for chemistry applications based on elementary symmetric polynomials was previously suggested by Opalka and Domcke as well as Schuurman, Neville and Wang in [52,75] based on classical Frobenius companion matrices. Our approach expands on these references in key aspects: First, we explore two natural alternative companion matrix approaches for this purpose – Schmeisser companion and Chebyshev colleague matrices – which have theoretical and practical advantages. In particular, using Frobenius companion matrices for root-finding problems can be unstable in a lot of practically important cases. The second key difference is that we provide a numerical analysis description of these approaches. The latter is of particular importance for the algorithm's usefulness in the actual chemistry machine learning applications where understanding sensitivity of the methods to perturbations in the underlying data has critical consequences for performance and thus feasibility. We present numerous numerical experiments which agree with the results predicted by our analysis and test them in the application-relevant high noise regimes.

2. Background

2.1. Elementary symmetric polynomials

A *symmetric* function f of m variables is a function that is invariant under permutations of its arguments, i.e.,

$$f(x_1, \ldots, x_m) = f(x_{\sigma 1}, \ldots, x_{\sigma m}) \qquad \forall \sigma \in S_m,$$

or $f \circ \sigma = f$ in short. Certain types of symmetric polynomials, such as of elementary, homogeneous, monomial and power sum types (cf. [40, Ch.9]), are distinguished by various convenient properties and have received much attention in the fields of algebraic geometry, combinatorics and beyond [16,40,65]. The numerical method we introduce in this paper relies primarily on properties of the elementary symmetric polynomials.

Definition 2.1. (Elementary symmetric polynomials, [16, Ch.2] & [40, Ch.9])
The k-th *elementary symmetric polynomial* (ESP) in m variables is defined by

$$s_k(x_1, \ldots, x_m) := \sum_{1 \le j_1 < j_2 < \cdots < j_k \le m} \prod_{i=1}^{k} x_{j_i}.$$

For example,

$$\begin{aligned}
s_0(x_1, x_2, \ldots, x_m) &= 1, \\
s_1(x_1, x_2, \ldots, x_m) &= \sum_{1 \le j \le m} x_j, \\
s_2(x_1, x_2, \ldots, x_m) &= \sum_{1 \le j < k \le m} x_j x_k, \\
&\vdots \\
s_m(x_1, x_2, \ldots, x_m) &= \prod_{j=1}^{m} x_j
\end{aligned}$$

and $s_k(x_1, x_2, \ldots, x_m) = 0$ for $k > m$.

Among the important properties of the elementary symmetric polynomials is that they establish a connection between the coefficients of a polynomial and its roots [23,27,72], commonly known as Viète's formula. We refer to [73, Section 3.2] for a proof.

Theorem 2.1 *(Viète's formula). Let $p(x) = x^n + \sum_{j=0}^{n-1} a_j x^j$ be a degree n monic polynomial with coefficients $a_j \in \mathbb{R}$ and (possibly complex and repeated) roots $r_j, j = 1, \ldots, n$, then*

$$s_k(r_1, \ldots, r_n) = (-1)^k a_{n-k}.$$

2.2. Companion matrices

In this section we review companion matrices of polynomials and some related constructions. Each square complex matrix A has an associated characteristic polynomial given by $p_A(\lambda) = \det(\lambda I - A)$, whose roots are precisely the eigenvalues of A [34,66]. Companion matrices [22,41,43] arise as the answer to the converse question: Given a polynomial $p(x)$, can one construct a matrix A_p whose eigenvalues are the roots, i.e. $p(x) = \det(xI - A_p)$? This question is particularly interesting for numerical computations since it allows one to replace root finding problems with well-understood eigenvalue problems [1,2,15,69].

Theorem 2.2. *(Frobenius companion matrix, [34, Ch.3]) Let $p(x) = x^n + \sum_{k=0}^{n-1} a_k x^k$ be a degree n monic polynomial. We define the* companion matrix A_p^{F} *of $p(x)$ as*

$$A_p^{\mathrm{F}} = \begin{pmatrix} 0 & 0 & \ldots & 0 & -a_0 \\ 1 & 0 & \ldots & 0 & -a_1 \\ 0 & 1 & \ldots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & -a_{n-1} \end{pmatrix}.$$

Then, the eigenvalues of the companion matrix A_p^{F} are the roots of $p(x)$.

Since matrix eigenvalues are preserved under similarity transforms, the Frobenius companion matrix A_p^{F} is not a unique solution to $p(x) = \det(xI - A)$; see, e.g., the pentadiagonal Fiedler companion matrices [19,20,44]. Our work uses Schmeisser's related construction of symmetric tridiagonal companian matrices [60].

Theorem 2.3 *(Schmeisser companion matrix, [60]). Let $p(x) = x^n + \sum_{k=0}^{n-1} a_k x^k$, $a_j \in \mathbb{R}$ be a monic polynomial of degree n with only real (but not necessarily distinct) roots. Then there exists a real symmetric tridiagonal matrix A_p^{S} with non-negative off-diagonal entries*

$$A_p^{\mathrm{S}} = \begin{pmatrix} -q_1(0) & \sqrt{c_1} & 0 & \ldots & 0 \\ \sqrt{c_1} & -q_2(0) & \sqrt{c_2} & \ddots & 0 \\ 0 & \sqrt{c_2} & -q_3(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \sqrt{c_{n-1}} \\ 0 & \ldots & 0 & \sqrt{c_{n-1}} & -q_n(0) \end{pmatrix}, \tag{2.1}$$

for which $p(x)$ is the characteristic polynomial, that is

$$p(x) = \det\bigl(xI - A_p^{\mathrm{S}}\bigr).$$

Furthermore, Algorithm 1 provides a constructive method to obtain the entries of A_p^{S}.

Algorithm 1: Construction of sym. tridiagonal Schmeisser companion matrix [60].

Input: Monic polynomial $p(x) = x^n + a_{n-1}x^{n-1} + ... + a_0$, $a_j \in \mathbb{R}$, with exclusively real roots of arbitrary multiplicity. Coefficient vectors sorted by ascending order, i.e. $\mathbf{a} = (a_0, ..., a_{n-1}, 1)$ and indexed by 1 : end.
Output: Diagonal and off-diagonal entries (q, c) of A_p^{S} in Theorem 2.3.

```
Set y_1 ← a and y_2 ← a'/n where a' are the coefficients of p'(x).
for k = 1 to n do
    (u, r) ← Quotient and remainder of polynomial division of y_1 by y_2.
    if k < n then
        y_1 ← y_2
        y_2 ← r / r[end]
        c[k] ← −r[end]
    end
    q[k] ← −u[1]
end
return (q, c)
```

An attractive property of Schmeisser's construction is that the eigenvalues of symmetric tridiagonal matrices are not only guaranteed to be real but there are also efficient and well-understood numerical algorithms for the computation of these eigenvalues which guarantee real results, cf. [28,53].

2.3. Colleague matrices and Chebyshev polynomials

The final concept to review are the *colleague* matrices, which first require a brief discussion of root-finding with Chebyshev polynomials. It is a classical question how polynomials behave under perturbations of their coefficients [24,25,48,83,84]. A first cautionary tale is immediately found in x^2, where perturbation to $x^2+\epsilon$ by a small $0 < \epsilon \ll 1$ changes the roots from a multiplicity two real root at 0 to entirely imaginary roots $\pm i\sqrt{\epsilon}$ with the error scaling with $\sqrt{\epsilon} \gg \epsilon$. In this case the multiplicity causes the issue but as was shown by Wilkinson in [80] this is not always the case: Wilkinson's example polynomial $\prod_{n=1}^{20}(x-n)$ has exclusively real and distinct roots $\{1, 2, ..., 20\}$ but when expanded into a monomial series has extremely poor numerical root-finding conditioning, cf. [48] and [70, Part 3].

Despite these observations, polynomial root-finding is not necessarily ill-conditioned. As described in [69], using a companion matrix based root-finding method for a polynomial expressed in the monomial basis is well-behaved if the roots are near the complex unit circle but otherwise becomes ill-conditioned [63,68]. In general, the conditioning with respect to perturbed coefficients depends on (1) the multiplicities of the roots, (2) the basis in which a polynomial is expressed and (3) the location of its roots in the complex plane.

The Chebyshev polynomials of first kind $\{T_j(x)\}_{j\in\mathbb{N}_0}$ are a basis of polynomials orthogonal with respect to the following inner product:

$$\langle T_n, T_m \rangle_T = \int_{-1}^{1} T_n(x)\, T_m(x)\, \frac{1}{\sqrt{1-x^2}} \mathrm{d}x = \begin{cases} 0 & \text{if } n \neq m, \\ \pi & \text{if } n = m = 0, \\ \frac{\pi}{2} & \text{if } n = m \neq 0 \end{cases}$$

and have many appealing theoretical and numerical properties [11,56,68]. Importantly, the roots of $p(x) = \sum_{j=0}^{n} b_j T_j(x)$ are well-conditioned as a function of b_j if its roots lie on or near $[-1, 1]$ and an eigensolver for the matrix introduced in the following theorem is used [29,49,50,62,64,68].

Theorem 2.4 *(Colleague matrix, [29,68]). The roots of the polynomial* $p(x) = T_n(x) + \sum_{j=0}^{n-1} b_j T_j(x)$, $b_j \in \mathbb{R}$, *are the eigenvalues of the following matrix which is commonly called the* colleague matrix*:*

$$A_p^{\mathrm{C}} = H - \frac{1}{2} e_1 c^\top = \begin{pmatrix} 0 & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ & \frac{1}{2} & \ddots & \ddots & \\ & & \ddots & 0 & \frac{\sqrt{2}}{2} \\ & & & \frac{\sqrt{2}}{2} & 0 \end{pmatrix} - \frac{1}{2} e_1 \begin{pmatrix} b_{n-1} & \cdots & b_1 & \sqrt{2} b_0 \end{pmatrix}, \quad (2.2)$$

Remark 2.1. The colleague matrix A_p^{C} in Theorem 2.4 is an upper Hessenberg matrix and is expressed as a sum of a real symmetric matrix and a rank-1 perturbation.

An overview of numerical algorithms, as well as a provably component-wise backward stable $O(n^2)$ QR algorithm for the diagonalization of colleague matrices was recently given by Serkh and Rokhlin [62].

3. Reconstruction of multi-surfaces

3.1. Problem statement and preliminaries

We assume that a union of $m \geq 2$ (hyper-)surfaces is given as graphs over some domain $\Omega \subseteq \mathbb{R}^d$ and call such a collection a *multi-surface* for simplicity. In most applications the relevant range of m is between 2 and 100 but there are some where even higher m would be of interest. The methods we develop in the present work are in principle general, but in practice will likely be directly applicable only to relatively few surfaces (say, up to 10).

Corresponding with our applications, we assume that at some finite collection of points $x \in \Omega \subseteq \mathbb{R}^d$ we can evaluate the surfaces sorted by their values. Any ordering can always be sorted to match this scenario and thus this comes at no loss of generality. Notably this means that there will generally be non-smooth cusps in the j-th entry function of our data as shown in Fig. 1 causing challenges for globally smooth, e.g. polynomial, approximation approaches.

Our aim is to reconstruct the multi-surface from given value-ordered point-wise data using a globally smooth interpolation method with good numerical properties (convergence, error control, etc.). This goal is motivated in large part by the use of machine learning approaches in the intended applications which rely on learning globally smooth interpolations. The following theorem provides the rigorous justification for reconstructing the multi-surfaces in this way as opposed to simply approximating the non-smooth surfaces featuring cusps, cf. Fig. 1, directly.

Theorem 3.1. *[68, Theorem 7.2] For $\nu \in \mathbb{N}$, let a function f and its derivatives up to $f^{(\nu-1)}$ be absolutely continuous on $[-1,1]$ and let $f^{(\nu)}$ satisfy $V = \|f^{(\nu+1)}\|_1 < \infty$ (bounded variation). Then for any $n > \nu$, the degree n Chebyshev interpolation of f denoted p_n satisfies*

$$\|f - p_n\|_\infty \leq \frac{4V}{\nu(n-\nu)^\nu} = O(n^{-\nu}).$$

If f is analytic in $[-1,1]$ then there exist constants $c, \alpha > 0$ such that

$$\|f - p_n\|_\infty \leq ce^{-\alpha n}.$$

The cusps seen in Fig. 1 correspond to the case $\nu = 1$ in Theorem 3.1 (when rescaled to $[-1,1]$), i.e. the cusp-containing surfaces are absolutely continuous but their derivatives which exist almost everywhere are not (though they have bounded variation). As a result, one should not expect convergence better than $O(n^{-1})$ when using a global Chebyshev approximation for the j-th entry multi-surface functions. One readily observes that if the ESPs are smooth then a Chebyshev interpolation of these functions converges exponentially in degree (interpolation point spacing and number, among other considerations, may put practical limitations on this theoretical behavior).

In the next sections we describe three closely related methods for the reconstruction of multi-surfaces from global interpolants of invariants. The three methods correspond, respectively, to employing Frobenius companion matrices, Schmeisser companion matrices and Chebyshev colleague matrices in the reconstruction, each with distinct conceptual or numerical advantages.

3.2. Reconstruction from invariants, Frobenius variant

We begin with the conceptually simplest variant using the classical Frobenius companion matrices and motivate the other variants as modifications thereof. We first describe the idea of the method for the simple case of three intersecting, value-sorted surfaces $\mathbf{f}(\mathbf{x}) = \mathrm{sort}(f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}))$, then present the general case methods. First, we construct the ESP values of the surfaces at each given point, i.e. for the case of three surfaces we have

$$
\begin{aligned}
s_1(\mathbf{f}(\mathbf{x})) &= \mathbf{f}(\mathbf{x})_1 + \mathbf{f}(\mathbf{x})_2 + \mathbf{f}(\mathbf{x})_3,\\
s_2(\mathbf{f}(\mathbf{x})) &= \mathbf{f}(\mathbf{x})_1\mathbf{f}(\mathbf{x})_2 + \mathbf{f}(\mathbf{x})_1\mathbf{f}(\mathbf{x})_3 + \mathbf{f}(\mathbf{x})_2\mathbf{f}(\mathbf{x})_3,\\
s_3(\mathbf{f}(\mathbf{x})) &= \mathbf{f}(\mathbf{x})_1\mathbf{f}(\mathbf{x})_2\mathbf{f}(\mathbf{x})_3.
\end{aligned}
$$

In a typical application scenario where one cannot evaluate the surfaces at arbitrary points but must instead work with pre-existing data, one would fit a (typically global) approximation of the three ESPs using e.g. orthogonal polynomials. Note that the j-th element of $\mathbf{f}(\mathbf{x})$, denoted $\mathbf{f}(\mathbf{x})_j$, is generally *not* equal to $f_j(\mathbf{x})$ due to the lack of sorting, cf. Fig. 1. However, by construction the ESPs are invariant under permutation of their arguments, meaning that any information about the sorting applied to $\mathbf{f}(\mathbf{x})$ is lost in this process. By Viète' formula (Theorem 2.1), the values of $\mathbf{f}(\mathbf{x})_j$ can be recovered from the ESPs by finding the roots of the monic polynomial

$$
p(y) = y^3 - s_1(\mathbf{f}(\mathbf{x}))y^2 + s_2(\mathbf{f}(\mathbf{x}))y - s_3(\mathbf{f}(\mathbf{x})),
$$

which we can achieve by computing the eigenvalues of its 3×3 Frobenius companion matrix $A_p^{\mathrm{F}}(\mathbf{x})$ as defined in Section 2.2 at each point $\mathbf{x} \in \Omega$, with

$$
A_p^{\mathrm{F}}(\mathbf{x}) = \begin{pmatrix} 0 & 0 & s_3(\mathbf{f}(\mathbf{x})) \\ 1 & 0 & -s_2(\mathbf{f}(\mathbf{x})) \\ 0 & 1 & s_1(\mathbf{f}(\mathbf{x})) \end{pmatrix}.
$$

The order of the returned eigenvalues may differ depending on the solver, so to guarantee consistency we sort the obtained eigenvalues $\lambda_{A_p^{\mathrm{F}}(\mathbf{x})}$ by value to obtain

$$
\operatorname{sort}(\lambda_{A_p(\mathbf{x})}) = \mathbf{f}(\mathbf{x}) = \operatorname{sort}(f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})).
$$

In describing this method for the reconstruction of $\mathbf{f}(\mathbf{x})$ from its ESPs, it may appear as if we have walked in a circle and simply regained what we started from: We were given the value-sorted $\mathbf{f}(\mathbf{x})$, constructed the ESPs from it and recovered the value-sorted $\mathbf{f}(\mathbf{x})$ from the associated companion matrix eigenvalue problem. The important observation which turns this into a useful framework is that while $\mathbf{f}(\mathbf{x})$ has cusps and is thus not well-approximated by globally smooth approximations (e.g. orthogonal polynomials), the ESPs are guaranteed to be smooth if *any* underlying smooth ordering of $\mathbf{f}(\mathbf{x})$ exists (by permutation invariance and linearity). Furthermore, as we will explore for conical cusps in the numerical experiments in Section 5.1.2, the only condition for well-approximable ESPs is their own smoothness which may be given even in situations where the underlying surfaces are not smooth.

The general procedure for arbitrary number of surfaces is described in Algorithm 2, which is exact in exact arithmetic. As mentioned in the introduction, Algorithm 2 is conceptually closely related to ideas suggested in [52,75].

There are several drawbacks of the method presented in Algorithm 2, first and foremost the fact that near cusps (i.e. multiple roots of the underlying polynomial) the

Algorithm 2: Pointwise reconstruction from smooth interpolants via Frobenius companion matrix.

Input: Elementary symmetric polynomials $\mathbf{s}(\mathbf{f}(\mathbf{x}))$ corresponding to an underlying multi-surface $\mathbf{f}(\mathbf{x})$ with m entries at a given point $\mathbf{x}' \in \Omega \subset \mathbb{R}^d$.
Output: Pointwise values of multi-surface $\mathbf{f}(\mathbf{x}')$ sorted by value.
$A_p^{\mathrm{F}} \leftarrow$ Companion matrix with coefficients $(-1)^{m+1} s_m(\mathbf{f}(\mathbf{x}))$ via Theorem 2.2.
$\lambda \leftarrow \mathrm{eigenvalues}(A_p^{\mathrm{F}})$
return $\mathrm{sort}(\lambda)$

eigenvalue computation can produce incorrect complex roots due to numerical errors, which when interpreted in terms of their real or absolute values cause surfaces to clamp together instead of intersecting. Next, we discuss the Schmeisser variant which helps address some of the shortcomings.

3.3. Symmetric tridiagonal Schmeisser variant

In Algorithm 3 we present an algorithm which reconstructs pointwise multi-surface values from the globally smooth ESPs thereof using the Schmeisser companion matrix which due to its real symmetric nature is guaranteed to have real eigenvalues. If appropriate eigensolvers are used (denoted in Algorithm 3 as *eigh()* due to the NumPy convention), the numerical eigensolver is also guaranteed to produce real eigenvalues, making complex roots impossible – albeit with a caveat.

Algorithm 3: Pointwise reconstruction from smooth interpolants from the Schmeisser companion matrix.

Input: Elementary symmetric polynomials $\mathbf{s}(\mathbf{f}(\mathbf{x}))$ corresponding to an underlying multi-surface $\mathbf{f}(\mathbf{x})$ with m entries at a given point $\mathbf{x}' \in \Omega \subset \mathbb{R}^d$.
Output: Pointwise values of multi-surface $\mathbf{f}(\mathbf{x}')$ sorted by value (guaranteed real).
$A_p^{\mathrm{S}} \leftarrow$ Schmeisser companion matrix for $(-1)^{m+1} s_m(\mathbf{f}(\mathbf{x}))$ via Algorithm 1
$\lambda \leftarrow \mathrm{eigh}(A_p^{\mathrm{S}})$
return $\mathrm{sort}(\lambda)$

Analogous to the discussion in the previous section, Algorithm 3 is exact in exact arithmetic. However, if there is sufficient noise on the input, then it is possible that the polynomial with theoretically guaranteed real roots is perturbed into a polynomial with complex roots (cf. the discussion in Section 2.3 for monomial series). If the Schmeisser construction in Algorithm 3 is attempted with a polynomial whose roots are complex, then it will fail since a polynomial with complex roots cannot have a Hermitian companion matrix. In practice the failing occurs such that the c_j on the off-diagonals of the Schmeisser matrix (see the definition in Theorem 2.3) become negative due to noise or numerical round-off, causing the square root to be complex, resulting in a non-Hermitian matrix if complex square roots are allowed.

One way to avoid this is motivated by a classical result: Again analogous to the Frobenius method, the surface crossings occur precisely where two or more eigenvalues

are identical – more precisely, no surfaces cross if all eigenvalues of the companion matrix are *simple*. Unlike for Frobenius companion matrices, however, one can often determine based on the off-diagonal entries of the Schmeisser companion matrix whether two or more surfaces are close to each other or crossing due the following classical result for symmetric tridiagonal matrix eigenvalues:

Lemma 3.2. *([53, Section 7]) Let A be a real symmetric tridiagonal matrix. If all the off-diagonal entries are positive, then all eigenvalues of A are simple.*

Note that this implies that crossings may be close if an off-diagonal term c_j of the Schmeisser companion matrix vanishes or approximately vanishes but the converse is not necessarily true as Wilkinson provided examples now known as Wilkinson matrices [81] which are symmetric, tridiagonal and have two very close but not exactly identical eigenvalues while having super and subdiagonal entries far removed from 0. Throughout numerous numerical experiments, however, no Wilkinson-like matrices were observed when constructing Schmeisser's companion matrix.

To avoid small negative values in the c_j from erroring in numerical implementations, one could introduce a filtering step into Algorithm 3, e.g. $c_j \leftarrow \max(0, c_j)$, effectively corresponding to a manual clamping of values or regularize the computation in one of many ways, corresponding to enforcing a minimum non-zero gap size. Beyond Wilkinson's counterexamples, one must also consider that for any $\epsilon > 0$ one chooses, surfaces can be constructed which come closer to touching than ϵ without in fact crossing. Post-processing approaches like this must thus be used with care and be adapted to the requirements of a specific application.

3.4. Chebyshev colleague matrix variant

A Chebyshev colleague matrix variant of Algorithms 2 and 3 is straightforward to construct as seen in Algorithm 4 but requires access to Chebyshev coefficients b_j such that

$$T_m(y) + \sum_{j=0}^{m-1} b_j(\mathbf{x}')T_j(y) = \sum_{k=0}^{m}(-1)^{m-k}s_{m-k}(\mathbf{f}(\mathbf{x}'))y^k. \tag{3.1}$$

Generically, the Chebyshev coefficients of any sufficiently regular function f can be obtained by the Chebyshev inner product (cf. [38, Ch.4])

$$b_j = \frac{2}{\pi}\langle f, T_j\rangle_T = \int_{-1}^{1} f(x)\,T_j(x)\,\frac{1}{\sqrt{1-x^2}}\mathrm{d}x = \frac{2}{\pi}\int_0^{\pi} f(\cos(\theta))\cos(j\theta)\mathrm{d}\theta,$$

where such an integral can be computed numerically or analytically. While we do not know of a closed-form representation of the required Chebyshev coefficients for Viète

Table 1
Monic Chebyshev coefficients b_j for the order m Viete polynomial in (3.1).

	$m=2$	$m=3$	$m=4$
b_0	$1+2x_1x_2$	$-2(x_1+x_2+x_3+2x_1x_2x_3)$	$3+4x_3x_4+4x_2(x_3+x_4)$ $+4x_1(x_2+x_3+x_4+2x_2x_3x_4)$
b_1	$-2(x_1+x_2)$	$3+4x_2x_3+4x_1(x_2+x_3)$	$-2(3(x_3+x_4)+x_2(3+4x_3x_4)$ $+x_1(3+4x_3x_4+4x_2(x_3+x_4)))$
b_2	1	$-2(x_1+x_2+x_3)$	$4(1+x_3x_4+x_2(x_3+x_4)+x_1(x_2+x_3+x_4))$
b_3		1	$-2(x_1+x_2+x_3+x_4)$
b_4			1

polynomials, this poses no difficulty for our proposed algorithm as for any fixed number of intersecting surfaces m the coefficients can be symbolically pre-computed, e.g. using the following monomial conversion rule often used in the context of power sum economization [38,56,67].

Lemma 3.3 *([21]). Let $n \in \mathbb{N}_0$ and $\binom{n}{k}$ denote the binomial coefficient, then*

$$x^j = \sum_{k=0}^{j} \gamma_{j,k} T_k(x),$$

where

$$\gamma_{j,k} = \begin{cases} 2^{-j}\binom{j}{(j-k)/2}, & \text{if } (j-k) \text{ even and } k=0, \\ 2^{1-j}\binom{j}{(j-k)/2}, & \text{if } (j-k) \text{ even and } k\neq 0, \\ 0, & \text{else.} \end{cases}$$

Alternatively, one can also use special case versions of Salzer's algorithm [30,59]. Using such procedures to compute Chebyshev coefficients is typically ill-advised in the context of function approximations since it involves first computing a non-Chebyshev series which can lead to loss of many of the Chebyshev polynomials' advantageous approximation properties and often requires higher precision arithmetic for sensible results, cf. [10]. However, in our method the degree of the polynomial is always fixed to the number of intersecting surfaces m, meaning that the required form of the Chebyshev linear combinations of the ESPs can be symbolically pre-computed and then evaluated either directly or with certain backward stability guarantees using e.g. Clenshaw's algorithm [9] (cf. [4,17]), instead of numerically computing the ESPs first. In Table 1 we provide a list of the first few coefficients computed using the general Wolfram Mathematica script in Appendix A which can be used to pre-compute them for arbitrary m. In general, if stability guarantees are required for this aspect of the algorithm, the most straightforward approach is to use the known symbolic forms and proceed with this evaluation symbolically in parallel, though this was not done for any of the numerical experiments we conducted.

Algorithm 4: Pointwise reconstruction from smooth interpolants from the Chebyshev colleague matrix.

Input: Chebyshev coefficients $b_j(\mathbf{x}')$ such that

$$T_m(y) + \sum_{j=0}^{m-1} b_j(\mathbf{x}')T_j(y) = \kappa \sum_{k=0}^{m} (-1)^{m-k} s_{m-k}(\mathbf{f}(\mathbf{x}'))y^k$$

with arbitrary $\kappa \in \mathbb{R} \setminus \{0\}$ for an underlying multi-surface $\mathbf{f}(\mathbf{x})$ with m entries at a given point $\mathbf{x}' \in \Omega \subset \mathbb{R}^d$.

Output: Pointwise values of multi-surface $\mathbf{f}(\mathbf{x}')$ sorted by value.

1 $A_p^{\mathrm{C}} \leftarrow$ Colleague matrix of b_j via Theorem 2.4

2 $\lambda \leftarrow \mathrm{eig}(A_p^{\mathrm{C}})$

3 **return** $\mathrm{sort}(\lambda)$

4. Sensitivity analysis

The chain of error propagation for our method is generally as follows in the most immediate use cases:

1. Point-wise value-sorted data is generated for a multi-surface $\mathbf{f}(\mathbf{x})$ subject to some noise ϵ_j on each entry, i.e. $\tilde{\mathbf{f}}(\mathbf{x})_j = \mathbf{f}(\mathbf{x})_j \pm \epsilon_j$. The nature of the noise is naturally application and implementation dependent.
2. From this data we compute the corresponding point-wise ESP or Chebyshev coefficient values as described in Section 3.4. This process is essentially linear combination and in the worst case scenario results in data with errors on the order $\epsilon = (m+1)\max_j \epsilon_j$.
3. An interpolation of the ESPs or Chebyshev coefficients b_j is computed. The error propagation into this step is sensitive to various aspects of the interpolation, including the interpolation degree used as well as the spacing of the data. Considering the 1D case for simplicity, if Chebyshev nodes are used for a Chebyshev interpolation then by Theorem 3.1 the error is expected to be

$$\|\mathbf{s}(\mathbf{f}(\mathbf{x}))_j - p_n\|_\infty \leq \|\mathbf{s}(\mathbf{f}(\mathbf{x}))_j - \mathbf{s}(\tilde{\mathbf{f}}(\mathbf{x}))_j\|_\infty + \|\mathbf{s}(\tilde{\mathbf{f}}(\mathbf{x}))_j - p_n\|_\infty = O(\epsilon) + O(e^{-\alpha n})$$

   if the conditions of that Theorem are satisfied by the perturbed invariants.
4. From the interpolated perturbed invariants $p_n \approx \mathbf{s}(\tilde{\mathbf{f}}(\mathbf{x}))_j$, we reconstruct the desired multi-surface points using a companion matrix approach.

We now investigate properties of the final step of the above sequence. An important result for understanding the sensitivity of our reconstruction methods to perturbations in exact arithmetic is the following classical theorem on the conditioning of polynomial root-finding in Chebyshev and monomial power bases.

Theorem 4.1. *[7, Theorem 4.1] Let r denote a real-valued root of multiplicity ℓ on the interval $x \in [-1,1]$ of a polynomial*

$$p(x) = \sum_{j=0}^{n} b_j \phi_j(x), \quad x \in [-1,1],$$

where either $\phi_j(x) = T_j(x)$ or $\phi_j(x) = x^j$, both of which satisfy $|\phi_j(x)| \leq 1$ for all $x \in [-1,1]$. Furthermore, let $\tilde{r}$ denote the root of the perturbed polynomial $\tilde{p}(x)$ with modified k-th coefficient:

$$\tilde{p}(x) = (b_k \pm \epsilon)\phi_k(x) + \sum_{j=0, j\neq k}^{n} b_j \phi_j(x), \quad x \in [-1,1].$$

Then, the shift of the root caused by the perturbation $0 < \epsilon \ll 1$ is bounded by

$$|r - \tilde{r}| \leq \epsilon^{\frac{1}{\ell}} \left| \frac{1}{\ell!} \frac{d^\ell p}{dx^\ell}(r) \right|^{-\frac{1}{\ell}} + O\left(\epsilon^{\frac{\ell+1}{\ell}}\right).$$

Abstractly, without taking numerical errors during the root-finding into account, Theorem 4.1 tells us that if $|\frac{1}{\ell!}\frac{d^\ell p}{dx^\ell}(r)|^{-\frac{1}{\ell}}$ is negligible our method will have errors of order $\sqrt{\epsilon}$ where two surfaces cross and of order ϵ where no surfaces cross, where ϵ is the local absolute error in our interpolation of the ESPs or Chebyshev coefficients as seen e.g. in Table 1. However, in general the applicability of the condition $|\epsilon| \ll 1$ strongly depends on the value of the relevant derivatives near the root. In the intended application of $\ell \in \{1,2\}$ the additional term represents the reciprocal quantity of the first and second derivatives near the given root which intuitively means that for $\ell = 1$ the error can be made arbitrarily large the closer a polynomial is to being constant near its root (or the slower its derivative changes for $\ell = 2$). An illustrative example provided by one of the anonymous reviewers is found in the roots of $p_1(x) = x^9 - 10^{-18}$ and $p_2(x) = x^9 + 10^{-18}$. Despite the perturbation being only of order 10^{-18} the root perturbations are at least of order 10^{-3} and inspection of $|\frac{dp_1}{dx}(r)|^{-1}$ reveals that it is of order 10^{15}. Since this problem requires specific circumstances to occur we have not observed this source of error in practice in our application-oriented numerical examples, though we nevertheless highlight that inspection of the derivatives near crossings in particular is essential and should be included in implementations to flag potentially unstable zones with need for better data or bespoke treatment as part of a piece-wise method. Note for this purpose that the derivatives of the Viète formula polynomial in Theorem 2.1 are readily accessible given the point-wise constant ESP coefficients.

Theorem 4.1 further shows that the errors of distinct roots are not coupled in well-behaved cases, i.e. are not affected by multiplicities other than their own. This means that there is no pollution to the other surfaces at points where two surfaces cross.

In the following sections we collect further results which can be used to understand the error incurred from the Schmeisser and Chebyshev colleague reconstruction approaches.

The sensitivity analysis necessarily differs between these methods as the Schmeisser matrix is something we have to construct using polynomial division, cf. Algorithm 1, which is an equivalent procedure to deconvolution. Since deconvolution is widely understood to be a numerically unstable process, we will focus the stability analysis for the Schmeisser companion matrix on perturbations on the companion matrices themselves, where meaningful error bounds can be given. In a practical application, however, care must be taken that sufficient precision is available for the polynomial division to proceed which may be done by using higher precision arithmetic, symbolic calculations or pre-processing steps as appropriate. In contrast, for the Chebyshev colleague matrix approach a classical result allows us to consider both perturbations to the ESP coefficients and the colleague matrix simultaneously, and thus effectively fully characterize the sensitivity to noise in the data.

4.1. Errors in the Chebyshev colleague matrix method

The Chebyshev colleague matrix approach shares a key strength with the Frobenius approach in that once the coefficients $b_j(\mathbf{x})$ discussed in Theorem 2.4 are computed there is no further error accumulation in the construction of the colleague matrix. A major drawback compared to the Schmeisser approach, however, is that the colleague matrix is in general not symmetric and thus the algorithm cannot guarantee real eigenvalues being returned for sufficiently large perturbations. To understand the sensitivity of the Chebyshev colleague reconstruction method we can leverage results for the error control of Chebyshev series root-finding.

Theorem 4.2. *[51, Corollary 5.4] Let $A_p^{\mathrm{C}} = H - e_1 c^\top$ with $H = H^\top$ be the Chebyshev colleague matrix of a polynomial $p(x) = T_n(x) + \sum_{j=0}^{n-1} b_j T_j(x)$ with $b_j \in \mathbb{R}$ as in Theorem 2.4 and $||\delta H||_2 \leq \epsilon_H$, $||\delta e_1||_2 \leq \epsilon_1$ and $||\delta c||_2 \leq \epsilon_c$. Then the perturbed matrix*

$$A_p^{\mathrm{C}} + \delta A_p^{\mathrm{C}} := H + \delta H - \frac{1}{2}(e_1 + \delta e_1)(c + \delta c)^\top$$

is the colleague matrix corresponding to a polynomial

$$p(x) + \delta p(x) = \sum_{j=0}^{n} (b_j + \delta b_j) T_j(x),$$

with perturbed coefficients $b_j + \delta b_j$, for which we have the bounds

$$|\delta b_j| \leq (6\|c\|_2 \epsilon_1 + 2\sqrt{n}\epsilon_c + (5 + 16\sqrt{n}\|c\|_2)\epsilon_H) n^2 + \mathcal{O}(\epsilon_H^2 + \epsilon_1^2 + \epsilon_c^2).$$

Remark 4.1. In the context of our problem, perturbations occur at the level of the data and thus the ESPs to be interpolated. The perturbations ϵ_1 and ϵ_H are of the order of numerical precision and can normally be neglected. The bound from Theorem 4.2 then simply reads

$$|\delta b_j| \leq 2n^{5/2}\|\delta c\|_2 + C\|\delta c\|_2^2,$$

where C may depend on $\|c\|_2$ and n.

If a backward stable algorithm is used to compute the eigenvalues of the colleague matrix, then the root-finding is backward stable as a function of perturbations of the Chebyshev series' coefficients, cf. [50, Cor. 2.8, Rem. 2.9] and [62]. Since no deconvolution is needed for the construction of the Chebyshev colleague matrix, the approach based on it is expected to perform well in generic circumstances.

4.2. Errors in the Schmeisser companion matrix method

Error bounds for the numerical eigenvalue problem associated with symmetric tridiagonal matrices have been a topic of long standing research [28,82]. More recent results [31–33] allow for the characterization of upper and lower bounds for the eigenvalues of symmetric tridiagonal interval matrices which are defined as follows.

Definition 4.1 *(Symmetric tridiagonal interval matrix).* We define the symmetric tridiagonal interval matrix $\mathbf{A} = [\underline{A}, \overline{A}]$ to be the set of all symmetric tridiagonal matrices A_j with entries taken from two given intervals, that is with all $a_j \in [\underline{a}_j, \overline{a}_j]$ with $j = 1, ..., n$ and all $b_j \in [\underline{b}_j, \overline{b}_j]$ with $j = 2, ..., n$.

We can think of interval matrices as representing a matrix with error bounds on each of its entries. These objects are studied in the field of interval arithmetic, providing an elegant way for rigorous numerics with guaranteed error bounds [46,47,71]. We seek error bounds on the eigenvalues, i.e. eigenvalue intervals $\boldsymbol{\lambda}_j = [\underline{\lambda}_j, \overline{\lambda}_j]$, of an interval matrix $\mathbf{A}$. Note that any method to determine the upper bounds $\overline{\lambda}_j$ could be used to determine the lower bounds via the transformation $A \mapsto -A$, cf. [31,32]. The following result, stated without proof in [57] and proved in [32], establishes a key result.

Theorem 4.3. *[32, Theorem 3.1] Let $\boldsymbol{\lambda}_j(\mathbf{A})$ denote the j-th eigenvalue interval of the symmetric tridiagonal interval matrix $\mathbf{A} = [\underline{A}, \overline{A}]$. We denote by A_c and A_Δ the mid point and radius matrices of the interval matrix defined by*

$$A_c := \frac{1}{2}\left(\overline{A} + \underline{A}\right), \qquad A_\Delta := \frac{1}{2}\left(\overline{A} - \underline{A}\right).$$

Then

$$\boldsymbol{\lambda}_j(\mathbf{A}) \subseteq [\lambda_j(A_c) - \rho(A_\Delta), \lambda_j(A_c) + \rho(A_\Delta)],$$

where $\rho(A_\Delta) := \max\{|\lambda_1(A_\Delta)|, ..., |\lambda_n(A_\Delta)|\}$ denotes the spectral radius of A_Δ.

In the context of the Schmeisser method discussed in Section 3, the natural interpretation of the interval matrix $\mathbf{A} = [\underline{A}, \overline{A}]$ is as

$$\mathbf{A} = A \pm \mathcal{E}_c = [A - \mathcal{E}_c, A + \mathcal{E}_c],$$

where $\mathcal{E}_c$ is an upper bound for the absolute error accumulated in the construction of the Schmeisser companion matrix via Algorithm 1. Defined in this way, the radius matrix $A_\Delta = \mathcal{E}_c$ is simply an absolute error bound matrix. With this established, we can state error bounds for computing the eigenvalues of a Schmeisser companion matrix.

Theorem 4.4. *Let $A(\mathbf{x})$ be a symmetric tridiagonal Schmeisser companion matrix with real, non-negative entries at each point $\mathbf{x} \in \Omega \subset \mathbb{R}$ and let $\mathcal{E}_c$ be a symmetric tridiagonal Toeplitz matrix with $a = b = c = \varepsilon_c \geq 0$. Then for any family of matrices $\hat{A}(\mathbf{x}) \in [A(\mathbf{x}) - \mathcal{E}_c, A(\mathbf{x}) + \mathcal{E}_c]$ we have*

$$\lambda_j(\hat{A}(\mathbf{x})) \in \boldsymbol{\lambda}_j([A(\mathbf{x}) - \mathcal{E}_c, A(\mathbf{x}) + \mathcal{E}_c]) \subseteq [\lambda_j(A(\mathbf{x})) - 3\varepsilon_c, \lambda_j(A(\mathbf{x})) + 3\varepsilon_c]$$

and furthermore

$$\sup_{\mathbf{x}\in\Omega} |\lambda_j(A(\mathbf{x})) - \lambda_j(\hat{A}(\mathbf{x}))| \leq 3\varepsilon_c.$$

Proof. From Theorem 4.3 we have

$$\begin{aligned}\lambda_j(\hat{A}(\mathbf{x})) \in &\boldsymbol{\lambda}_j([A(\mathbf{x}) - \mathcal{E}_c, A(\mathbf{x}) + \mathcal{E}_c]) \\ &= [\lambda_j(A(\mathbf{x})) - \max_j\{|\lambda_j(\mathcal{E}_c)|\}, \lambda_j(A(\mathbf{x})) + \max_j\{|\lambda_j(\mathcal{E}_c)|\}]\end{aligned}$$

That $\max_j\{|\lambda_j(\mathcal{E}_c)|\} \leq 3\varepsilon_c$ then follows from the Gershgorin circle theorem [6,26] since $\mathcal{E}_c$ is a tridiagonal Toeplitz matrix with $a = b = c = \varepsilon_c \geq 0$. The authors wish to thank the anonymous reviewer who helped to substantially simplify a prior version of this proof by suggesting reduction to the Gershgorin circle theorem. □

Theorem 4.4 establishes that the method in Algorithm 3 converges uniformly as a function of perturbations on the elements of the Schmeisser companion matrix. It is important to remind ourselves that this unfortunately does not necessarily translate to well-behaved convergence in terms of the error in an approximation of the ESPs and underlying data, since the construction of the Schmeisser companion matrix involves the numerically ill-conditioned step of polynomial division, which is mathematically equivalent to a deconvolution.

There are thus two primary drawbacks to the Schmeisser approach: First, the construction algorithm will fail if it is started with a ESP polynomial perturbed in such a way that its roots are no longer strictly real (though this can be substantially alleviated) and second, while there are tight error bounds for the eigenvalue computation once the

Schmeisser matrix is constructed, the stability of the Schmeisser construction algorithm itself is in doubt due to the equivalence of polynomial division and deconvolution. It remains to be seen whether the latter shortcoming can be addressed without costly use of higher precision arithmetic in the construction process, e.g. by pre-processing steps.

5. Numerical experiments

In this section we illustrate the performance of the proposed reconstruction methods in various toy problems designed to showcase generic interesting behavior with respect to convergence and stability as well as two applications-oriented examples with more complex behavior. In the context of the relevant applications, the most widely used error concepts are those of mean absolute error (MAE) and root mean square error (RMSE) rather than the more strict maximum absolute error often encountered in the context of polynomial approximations. While very useful for discussing worst-case behavior in toy problems with full control of the data, maximum absolute error is much more sensitive to outliers and thus less useful when investigating the different error behaviors in the application-adjacent examples. On the other hand, MAE and RMSE risk masking the extent to which the direct reconstruction method performs poorly by averaging out the error over the rest of the domain.

Thus, while we still also present these more conventional errors, we will introduce a gap-weighted error to discuss the specific behavior of interest which is the resolution of surface crossings in a multi-surface. Denoting by

$$\Delta_{ij}\mathbf{f}(\mathbf{x}) := \mathbf{f}(\mathbf{x})_j - \mathbf{f}(\mathbf{x})_i,$$

the difference between the i-th and j-th components of a multisurface $\mathbf{f}$ at $\mathbf{x}$, we introduce the following gap-weighted error for approximate reconstructions $\tilde{\mathbf{f}}$

$$\mathrm{err}_\Delta(\mathbf{f}, \tilde{\mathbf{f}}) = \max_{x\in\Omega, i\neq j} \frac{|\Delta_{ij}\tilde{\mathbf{f}}(\mathbf{x}) - \Delta_{ij}\mathbf{f}(\mathbf{x})|}{\epsilon_W + |\Delta_{ij}\mathbf{f}(\mathbf{x})|} \tag{5.1}$$

where $\epsilon_W > 0$ is a suitably small parameter to prevent the denominator from reaching 0 at crossings which we interpret as a target accuracy. This error places additional weight on good approximations near cusps as $\Delta_{ij}\mathbf{f}(\mathbf{x}) \to 0$ as $\mathbf{x}$ approaches a crossing site. We note that err_Δ is invariant under permutations of the surfaces. For all our numerical experiments we choose $\epsilon_W = 5 \times 10^{-2}$.

5.1. Toy problems

5.1.1. Smooth intersecting sinusoid surfaces

We begin with the problem shown in the introduction in Fig. 1 featuring the following three intersecting sinusoid curves on $[0, 2]$:

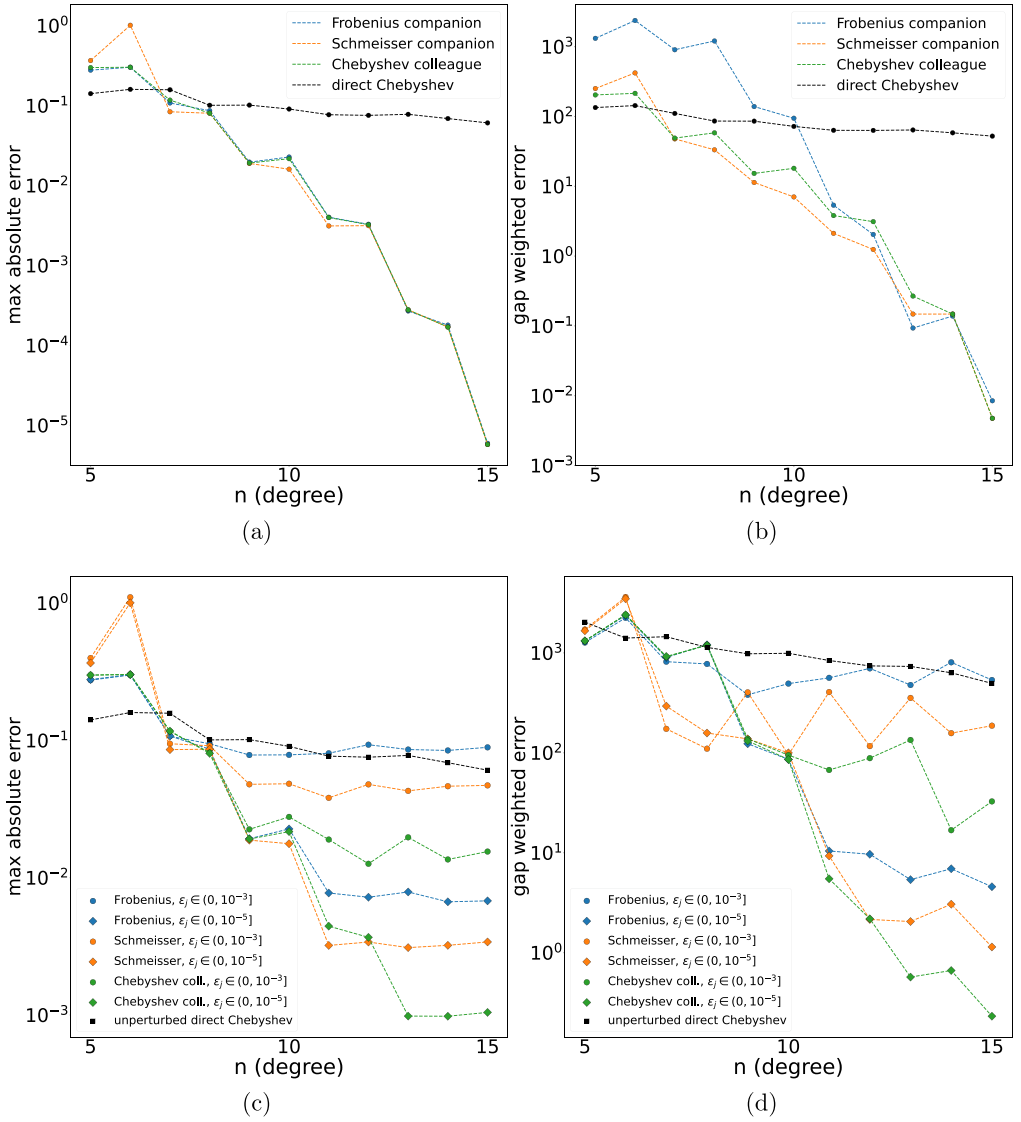


Fig. 2. (a-b) Semi-logarithmic maximum absolute error and gap weighted error on all 1000 random grid-points (sampled uniformly) for the problem in Eqs. (5.2)–(5.4) solved via Frobenius, Schmeisser companion, Chebyshev colleague or direct Chebyshev interpolation approaches. (c-d) show analogous error plots with added perturbations of indicated magnitude on the surface or ESP evaluations.

$$z_1(x) = \sin(x), \tag{5.2}$$

$$z_2(x) = \cos(2x), \tag{5.3}$$

$$z_3(x) = \sin(2x). \tag{5.4}$$

Fig. 2(a) shows maximum absolute error plots for the reconstruction using our proposed methods as well as a direct Chebyshev interpolation of the cusps as in Fig. 1(b) as

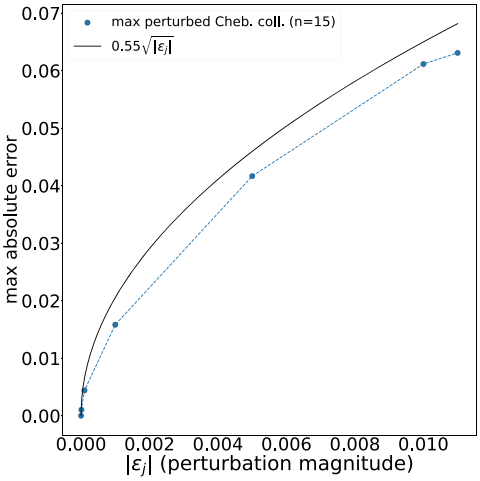


Fig. 3. Endpoint plot of Fig. 2(c) including some additional data points, showing that the converged (in degree) error for using the Chebyshev colleague approach for the problem in (5.2)–(5.4) scales like the square root of the perturbations, consistent with Theorem 4.1 in light of the multiplicity 2 roots observed in this problem, cf. Fig. 1.

a function of the interpolation degree using 1000 randomly distributed points as data (sampled from a uniform distribution), while Fig. 2(b) shows the gap weighted error defined in (5.1). As expected from the description given in Section 4, the companion matrix approaches achieve spectral convergence in degree given data saturation while attempting to simply resolve the cusps directly with Chebyshev polynomials results in approximately $O(n^{-1})$ convergence. Fig. 2(c) and (d) show analogous error plots but at each point the evaluation of the ESPs (and Chebyshev equivalents) are perturbed by the indicated different magnitude random amounts ϵ_j. We observe that in this more realistic scenario for machine learning applications, while the convergence rate is capped at a fixed amount, our methods nevertheless perform better once $\epsilon_j \lessapprox 0.001$ for the Schmeisser variant and $\epsilon_j \lessapprox 0.01$ for the Chebyshev colleague approach. Fig. 3 shows that the observed error behavior with respect to the magnitude of perturbations using the Chebyshev colleague method agrees with the prediction of Theorem 4.1.

As we are considering a toy problem with controlled uniform distribution perturbations without outliers or sampling bias related sources of error, the gap weighted error loosely retains the hierarchy seen in the max abs errors.

Next we consider the following three intersecting 2D surfaces (cf. Fig. 4):

$$\tilde{z}_1(x,y) = 2\sin\left(\tfrac{6(x+y)}{5}\right), \tag{5.5}$$

$$\tilde{z}_2(x,y) = \tfrac{2}{3} - \cos(x-y), \tag{5.6}$$

$$\tilde{z}_3(x,y) = 1. \tag{5.7}$$

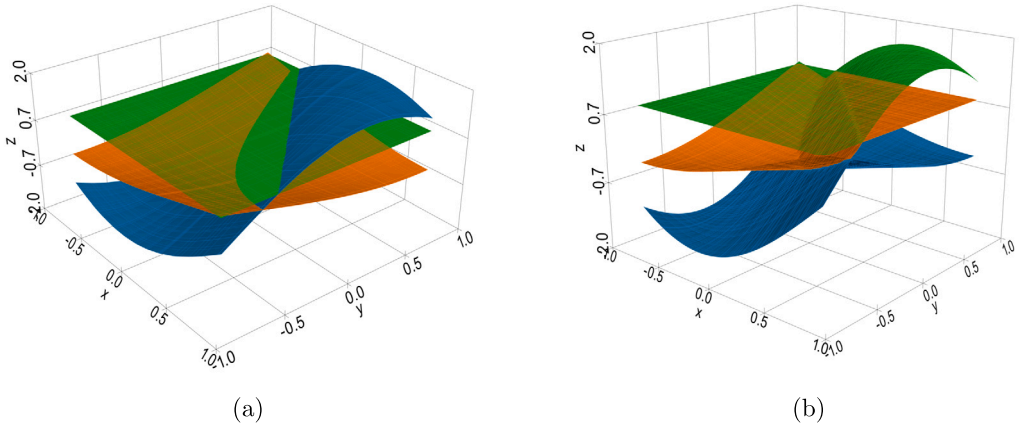


Fig. 4. (a) shows the surfaces $\tilde{\mathbf{z}}$ in (5.5)–(5.7), (b) shows a corresponding value-ordered reconstruction with cusps.

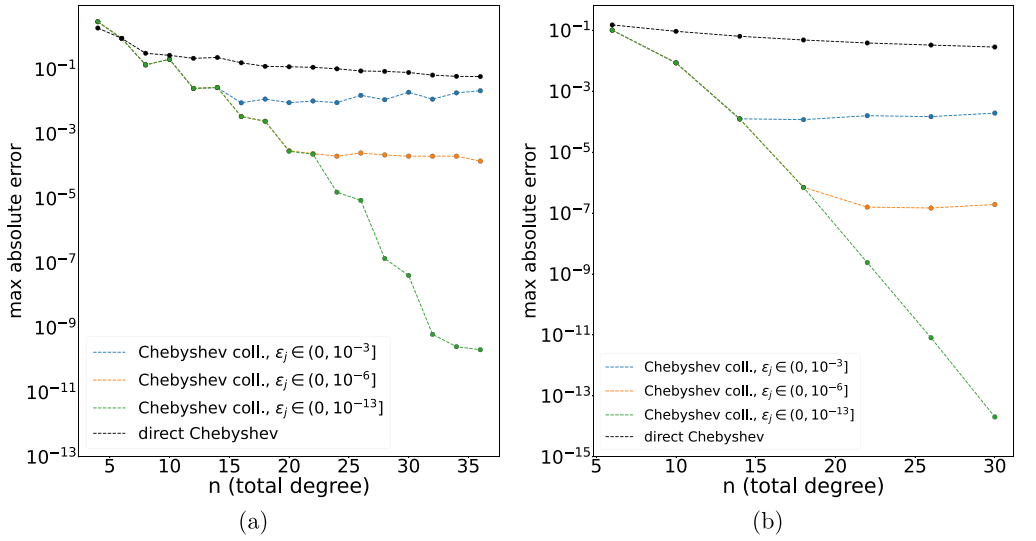


Fig. 5. (a) shows a semi-logarithmic plot of the max. abs. error for the problem in (5.5)–(5.7) using the Chebyshev colleague approach, (b) shows an analogous plot for reconstructing the conical cusp problem in (5.10)–(5.11).

We plot the max. abs. error of the direct reconstruction, Chebyshev colleague and Schmeisser companion approaches, including various levels of perturbation, in Fig. 5.

For the sake of presentation clarity we will henceforth mostly restrict our error plots to comparisons of Chebyshev colleague matrix and direct method reconstructions as the colleague matrix approach consistently performs either better or equivalently to Schmeisser or Frobenius companion matrix approaches without additional processing. Furthermore, we will only show max. abs. errors unless a meaningful qualitative difference in behavior could be observed from showing other types of errors.

5.1.2. Conical cusps in two dimensional surfaces

Let two 2D surfaces z_1 and z_2 be given as the two solutions of

$$z(x,y)^2 = \left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2,$$

with $0 \neq a, b \in \mathbb{R}$, i.e.

$$z_1(x,y) = \frac{\sqrt{a^2y^2+b^2x^2}}{ab}, \tag{5.8}$$

$$z_2(x,y) = -\frac{\sqrt{a^2y^2+b^2x^2}}{ab}. \tag{5.9}$$

While neither $z_1(x,y)$ nor $z_2(x,y)$ are even once continuously differentiable, it is easy to observe that their elementary symmetric polynomials are smooth:

$$\begin{aligned} s_1(\mathbf{z}(x,y)) &= z_1(x,y) + z_2(x,y) = 0, \\ s_2(\mathbf{z}(x,y)) &= z_1(x,y)z_2(x,y) = -\left(\tfrac{x}{a}\right)^2 - \left(\tfrac{y}{b}\right)^2. \end{aligned}$$

These surfaces describe the asymptotic behavior near conical cusps. Unfortunately the problem in (5.8)–(5.9) is too simple for a computational toy problem since s_1, s_2 are low degree polynomials and the reconstructions using companion matrix methods are thus almost immediately exact to their full precision. Thus, to make the problem more interesting, we consider a non-algebraic modification

$$\tilde{z}_1(x,y) = \sinh\left(\frac{\sqrt{a^2y^2+b^2x^2}}{ab}\right), \tag{5.10}$$

$$\tilde{z}_2(x,y) = \sinh\left(-\frac{\sqrt{a^2y^2+b^2x^2}}{ab}\right), \tag{5.11}$$

which also has a conical cusp at $(x,y) = (0,0)$ with smooth ESPs:

$$\begin{aligned} \tilde{s}_1(\tilde{\mathbf{z}}(x,y)) &= \tilde{z}_1(x,y) + \tilde{z}_2(x,y) = 0, \\ \tilde{s}_2(\tilde{\mathbf{z}}(x,y)) &= \tilde{z}_1(x,y)\tilde{z}_2(x,y) = -\sinh^2\left(\frac{\sqrt{a^2y^2+b^2x^2}}{ab}\right). \end{aligned}$$

We plot the surfaces $z_1(x,y), z_2(x,y)$ as well as the smooth ESP $\tilde{s}_2(\mathbf{z}(x,y))$ with $(a,b) = (\frac{4}{3}, \frac{12}{5})$ in Fig. 6 and the error of the Chebyshev colleague approach for both problems in Fig. 5(b), showing that the method performs well for conical cusps. In Section 5.2.1 we explore the electronic band structure of graphene which is an example of an application featuring multiple conical cusps.

5.1.3. Non-smooth ESPs

If all to-be-reconstructed surfaces are smooth then it is guaranteed that the ESPs as well as the Chebyshev equivalents will be smooth. Furthermore, we have seen in the previous section that for certain cases of interest, e.g. conical intersections, it is possible

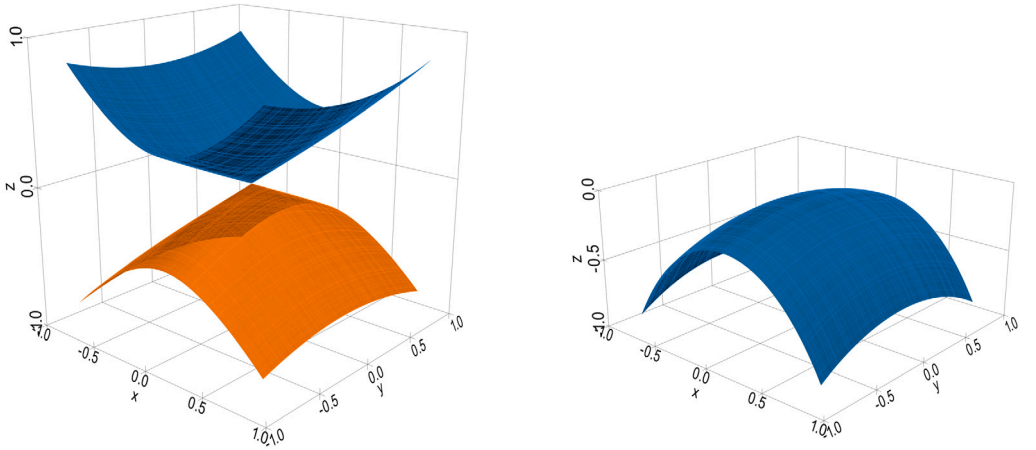


Fig. 6. (a) shows $\tilde{\mathbf{z}} = (\tilde{z}_1(x,y), \tilde{z}_2(x,y)$ in Eq. (5.10)–(5.11) for $(a,b) = (\frac{4}{3}, \frac{12}{5})$, (b) shows the corresponding smooth ESP $\tilde{s}_2(\mathbf{z}(x,y))$.

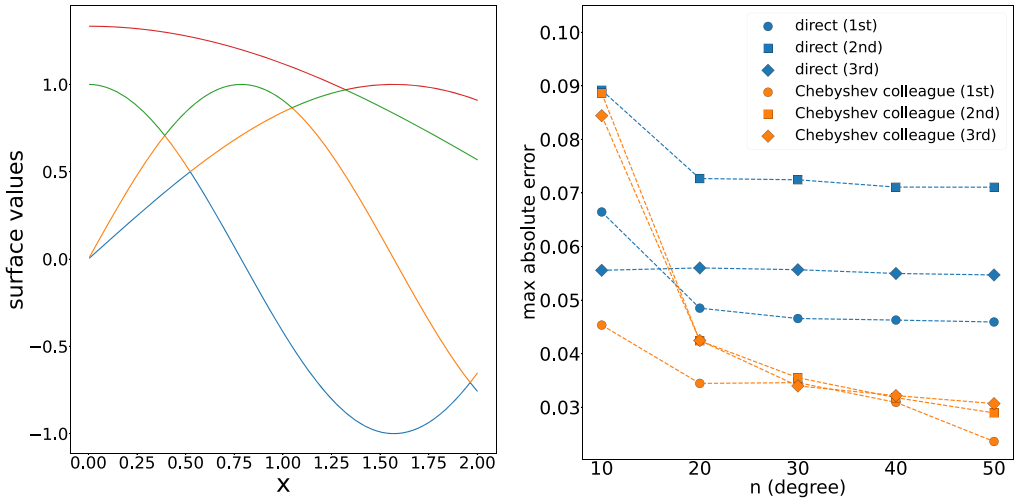
that the ESPs are smooth even if the underlying surfaces are not. In this section we consider a special case relevant for applications featuring non-smooth ESPs.

For simplicity we will consider a 1D example: We begin with a stack of m surfaces, ordered by their lowest attained value such that every surface intersects the next at least once in the domain of interest. We then consider the problem of reconstructing $m-1$ underlying surfaces from value ordered data such that at least one non-differentiable cusp remains in the highest value entry of our multi-surface data. Specifically, we will be considering the same problem from Section 5.1.1 and Fig. 1 but adding a fourth, not-to-be-reconstructed surface

$$z_4(x) = \tfrac{1}{3} + \cos\left(\tfrac{2x}{3}\right). \tag{5.12}$$

This scenario is significant for applications since e.g. the hierarchy of excited states involves in principle infinitely many stacked energy surfaces where one must choose an application-driven but ultimately arbitrary cutoff point.

We plot max. abs. errors for reconstructing the first three value-sorted surfaces described in (5.2)–(5.4) and (5.12) using direct interpolation and Chebyshev colleague matrix methods in Fig. 7(b). One observes that while the colleague matrix approach outperforms the direct method, the differences are less pronounced than in the case of smooth invariants. The observed behavior is expected since including a cusp in the invariants causes similar problems for the companion matrix approaches as it does for the direct method. The fact we still observe better errors is likewise expected since the number of cusps in the invariants is less than or equal to the number of unmatched / leftover cusps in the top surface times the number of surfaces to reconstruct, which is always less than or equal to the number of cusps in the original data. Informally, fewer cusps



(a) (b)

Fig. 7. (a) Intersecting, smooth sinusoidal curves in (5.2)–(5.4) and (5.12) pointwise sorted by values, cf. Fig. 1. Reconstructing only the lowest three layers leaves a cusp in the third layer near $x \approx 1.32$. (b) Linear scale max. abs. error plot for reconstructing the lowest three value-sorted surfaces from (a).

in the data lead to an expectation of better polynomial approximations, albeit only by constant factors. The companion matrix methods perform best in stacked surface scenarios if the cutoff surface is chosen such that the fewest possible number of cusps are left unmatched in the top layer. It is an interesting question whether there are practical ways to artificially match the leftover cusp in the top layer to restore smooth invariants.

5.2. Example applications

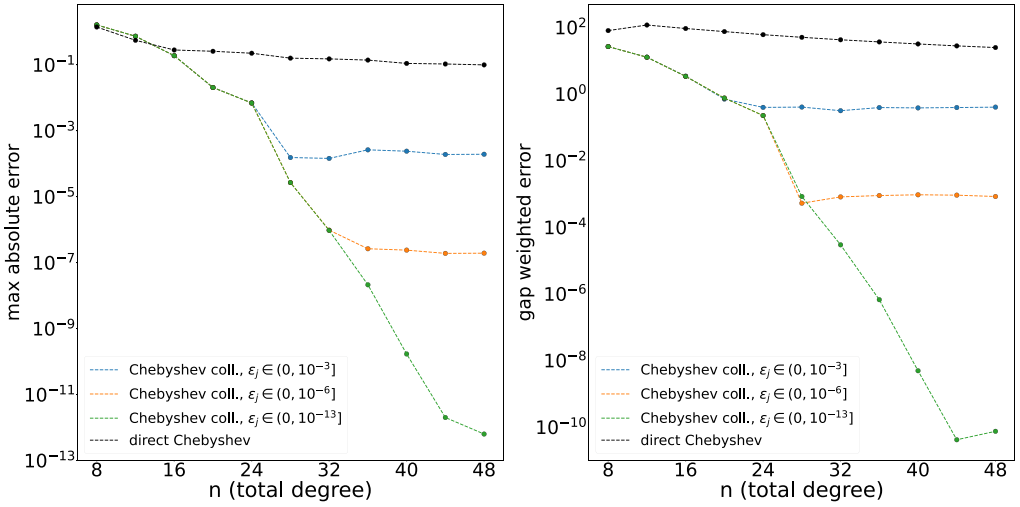
5.2.1. Conical cusps in the electronic band structure of graphene

Graphene consists of carbon atoms arranged in a hexagonal honeycomb lattice. For a single layer of graphene a tight-binding model (considering only nearest-neighbor interactions) can be used to obtain a Hamiltonian whose conduction and valance bands can be explicitly computed [39,61,74]:

$$E(\mathbf{k}) = \pm\gamma_0\sqrt{1 + 4\cos^2\left(\frac{ak_x}{2}\right) + 4\cos\left(\frac{ak_x}{2}\right)\cos\left(\frac{\sqrt{3}ak_y}{2}\right)}, \tag{5.13}$$

where $\mathbf{k} = [k_x, k_y]$ denotes the electron wave vector and $\gamma_0 \approx 2.8$ eV, $a \approx 2.46$Å are physical constants. For thorough discussions of the physics and chemistry of graphene we refer to [8,39] and the references therein.

As seen in Fig. 9(a), graphene's conduction and valence bands touch at the so-called Dirac points, making it a zero-gap semiconductor. In the language of the previous sections this means that the conduction-valence multi-surface of graphene has multiple conical



(a) (b)

Fig. 8. max. abs. error in (a) and gap weighted error in (b) on a 150×150 grid using the Chebyshev colleague matrix and direct interpolation methods with varying data perturbation ϵ_j for the Graphene example described in (5.13).

intersections. The electronic band structure of graphene thus constitutes a more advanced and application-oriented version of results discussed in Section 5.1.2. In Fig. 8 we plot the maximum absolute error for reconstructing the bands with various levels of perturbations prior to polynomial interpolation of the invariants.

In Fig. 9 we provide a visual comparison of the avoided crossings obtained using a direct interpolation of the surfaces versus those obtained via the Chebyshev colleague method, showing substantially better resolved Dirac cones. Fig. 9 also provides a clear demonstration of another key strength of this approach: Unlike a direct interpolation attempt, the presence of cusps does not pollute the rest of the multi-surface.

5.2.2. Energy surfaces of a sulfur dioxide (SO_2) molecule

In this section we use our method to reconstruct the energy surfaces for sulfur dioxide (SO_2), a molecule consisting of a sulfur and two oxygen atoms as pictured in Fig. 10, from global interpolants. We pre-compute (unordered) energies associated with different relative positions for different angles θ and distances d_1 and d_2 between the sulfur atom and respective oxygen atoms using a linear vibronic coupling (LVC) model of SO_2 [36, 55,76].

For larger molecules it would be natural to machine learn the ESPs from limited pre-computed samples using frameworks such as the atomic cluster expansion (ACE) [13,14] or MACE [5]. The example of SO_2 is sufficiently small, however, that we can treat the problem directly using Chebyshev polynomials since as seen in Fig. 10 the state is up to global symmetries fully determined from coordinate triplets such as (θ, d_1, d_2) or

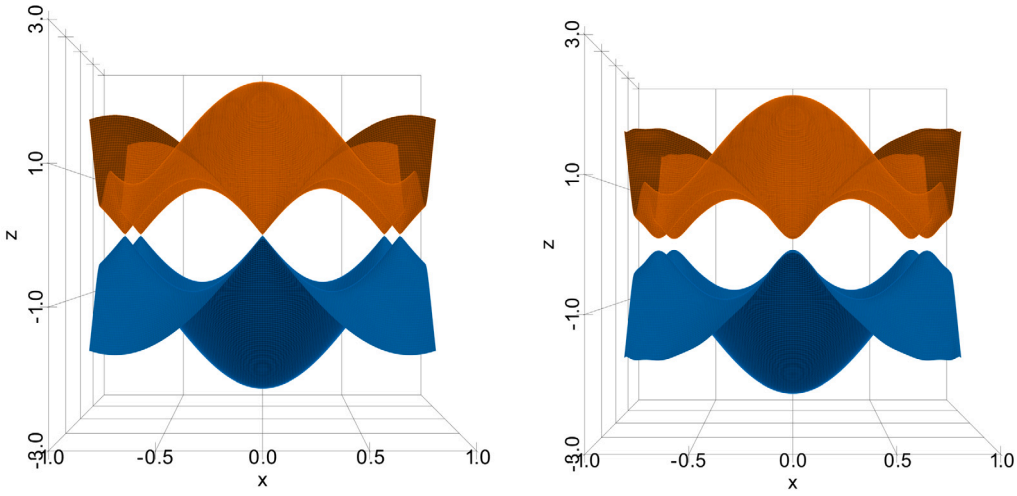


Fig. 9. (a) shows scaled, reconstructed Graphene conduction and valence bands using a Chebyshev colleague method (tot. degree $n = 40$) with the underlying data randomly perturbed by $\epsilon_j \in (0, 10^{-3}]$, (b) shows analogous reconstruction using direct approximation with Chebyshev polynomials (tot. degree $n = 40$) and exact data.

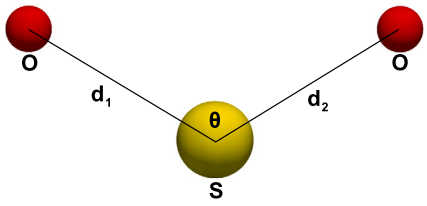


Fig. 10. Schematic of SO_2 molecule described by coordinate triplet (θ, d_1, d_2).

$(\theta, d_1 - d_2, d_1 + d_2)$. Since a conical intersection is known to occur in the potential energy surfaces for $d_1 - d_2 \approx 0$, cf. [79], we will use the latter set of internal coordinates.

As the energy surfaces for this setup are three-dimensional, a direct visualization is effectively impossible. We thus instead plot projected slices of our data in Fig. 12 showing the presence of a cusp. Since the underlying data is obtained from an LVC model, giving approximations to the actual electronic structure of the system, the results of this section are to be interpreted as relying on highly perturbed data of a non-negligible but unspecified amount similar to the scenario one would encounter in more complex computational chemistry applications. Errors due to outliers in the data dominate the max. abs. errors as well as overall MAE when reconstructing the surfaces, see Fig. 11(a), and thus as discussed at the beginning of Section 5 mask the improvement near the cusps plainly visible in Fig. 12. The gap weighted error defined in Eq. (5.1) which we plot in Fig. 11(b) provides a natural quantitative measure to distinguish between the cusp resolution quality of direct reconstructions and the Chebyshev colleague matrix approach.

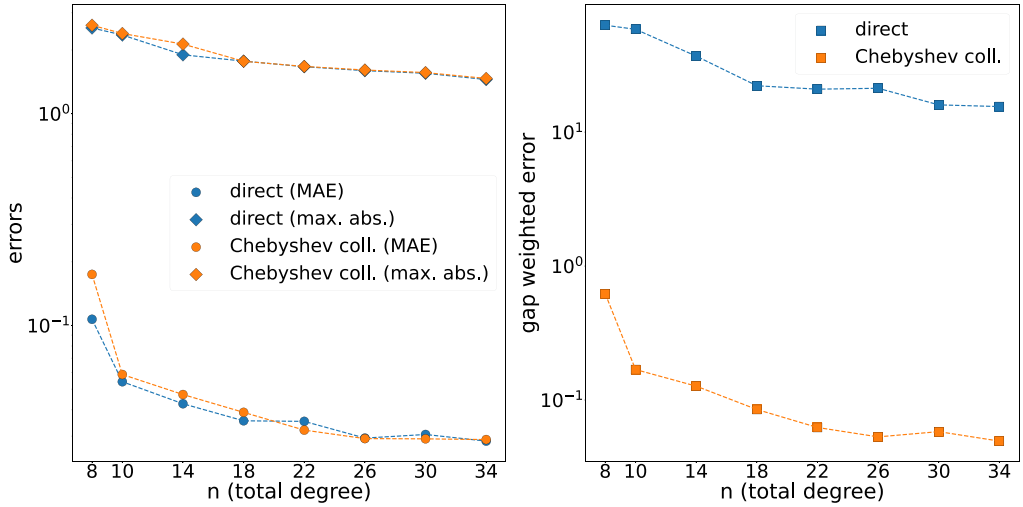


Fig. 11. (a) Semi-logarithmic error plot showing obtained MAE and max. abs. errors of the direct and Chebyshev colleague matrix methods associated with different configurations of SO_2, cf. Fig. 12. (b) is an analogous plot showing gap weighted errors.

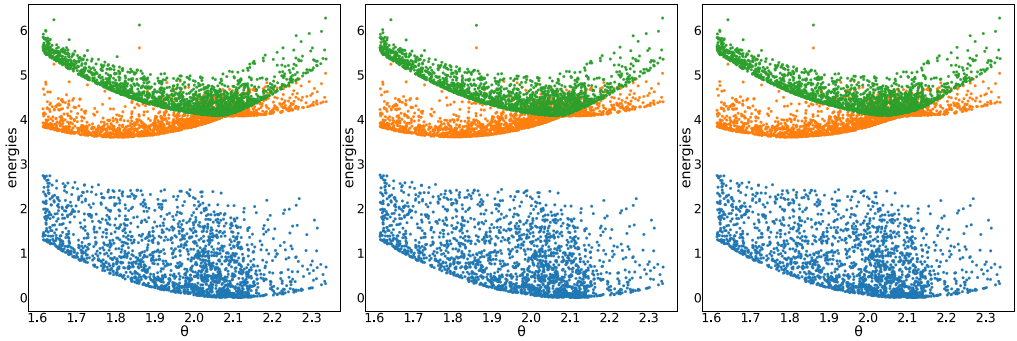
6. Discussion

We described a Schmeisser companion and Chebyshev colleague method to reconstruct a multi-surface from unordered or value-sorted data and explored their numerical properties. The methods have attractive approximation properties in low noise context and retain some of these properties into the more application-relevant noisy regimes. The primary expected value of these methods is improved resolution of multi-surface cusps with globally smooth interpolations or alternatively equivalent quality resolution of cusps at lower polynomial degree.

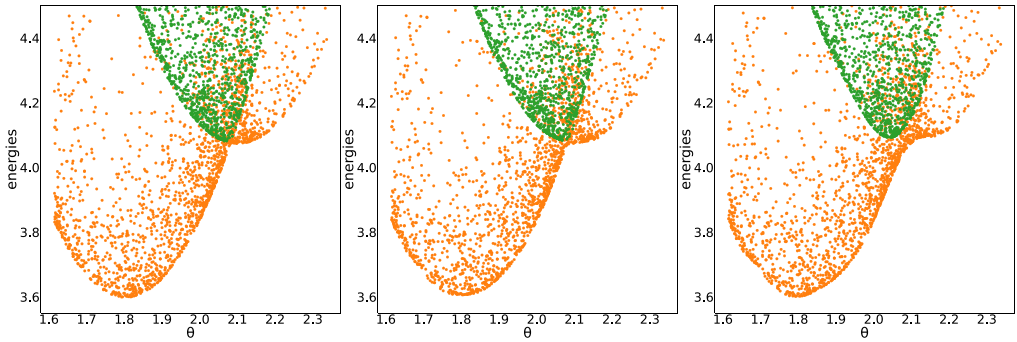
Throughout the paper we have mentioned opportunities to add pre- or post-processing steps in order to further improve the practical usefulness of these methods. An alternative approach could be the use of different smooth invariants altogether. A particularly interesting future direction would be the combination of such companion matrix based methods with machine learning models for molecular properties such as the atomic cluster expansion (ACE) [13,14] and related MACE [5] frameworks which rely on polynomial interpolations and are thus a natural fit.

Declaration of competing interest

There is no competing interest to declare.



(a) cusp in SO_2 data projection (b) Chebyshev colleague ($n = 28$) (c) direct method ($n = 28$)



(d) zoom in of (a) (e) zoom-in of (b) (f) zoom-in of (c)

Fig. 12. (a) shows cusp presence for a three-dimensional SO_2 dataset. The presented scatter plots are 1D θ-projections of 2D-slices with $d_1 - d_2 \approx 0$ aiming to visualize the presence of a cusp. (b) and (c) show reconstructions using Chebyshev colleague and direct methods at total degree $n = 28$. (d-f) show zoomed-in segments of (a-c).

Acknowledgements

This work was supported by NSERC Discovery Grant GR019381 and NFRF Exploration Grant GR022937. TSG was also supported by a PIMS-Simons postdoctoral fellowship, jointly funded by the Pacific Institute for the Mathematical Sciences (PIMS) and the Simons Foundation. This work was also supported by the Deutsche Forschungsgemeinschaft (DFG) – Project-ID 443871192 - GRK 2721: "Hydrogen Isotopes 1,2,3H." The authors would like to acknowledge the ZIH TU Dresden and the URZ Leipzig University for providing computational resources.

The authors acknowledge helpful conversations with Sebastian Mai and Cheuk Hin Ho on implementations, applications and motivation of the presented methods as well as Isaac Holt for feedback on the manuscript. Furthermore, the authors wish to thank the anonymous reviewer whose comments helped to significantly improve upon earlier versions of this paper.

Appendix A. Wolfram Mathematica script to pre-compute Chebyshev coefficients for Viète polynomials

The following script was tested in Wolfram Mathematica 13.3.1.0 [35]. The symbolic version of Salzer's algorithm is modified from a version published at [42].

```
In[1]:=
(* Returns the k-th ESP for n variables *)

xList[k_]:="x"<>ToString[#]&/@Range[k]

ESP[k_,n_]:=SymmetricPolynomial[k,xList[n]]

(* Compute Viete polynomial *)

VietePolynomial[m_] := Sum[(-1)^(m-k)*ESP[m-k,m]*λ^k,{k,0,m}]


In[4]:=
(* Symbolic version of Salzer's algorithm *)

ComputeCoefficients[poly_]:=Module[{c,n,a},

c=CoefficientList[poly,λ]; n=Length[c]-1; Remove[a];

a[0,2]=c[[n-1]]+c[[n+1]]/2; a[1,2]=c[[n]]; a[2,2]=c[[n+1]]/2;

Do[a[0,k+1]=c[[n-k]]+a[1,k]/2;

a[1,k+1]=a[0,k]+a[2,k]/2;

Do[a[m,k+1]=(a[m+1,k]+a[m-1,k])/2,{m,2,k-1}];

a[k,k+1]=a[k-1,k]/2;

a[k+1,k+1]=a[k,k]/2,{k,2,n-1}];

Table[a[m,n]/a[n,n],{m,0,n}]];
```

After initializing the above functions and fixing a given m, the command `Table[ComputeCoefficients[VietePolynomial[k]],k,2,m]//TableForm` will output a table like Table 1.

Data availability

Data will be made available on request.

References

[1] J.L. Aurentz, et al., Fast and backward stable computation of roots of polynomials, SIAM J. Matrix Anal. Appl. 36 (3) (2015).
[2] J.L. Aurentz, et al., Fast and backward stable computation of roots of polynomials, part II: backward error analysis, companion matrix and companion pencil, SIAM J. Matrix Anal. Appl. 39 (3) (2018).
[3] S. Axelrod, E. Shakhnovich, R. Gómez-Bombarelli, Excited state non-adiabatic dynamics of large photoswitchable molecules using a chemically transferable machine learning potential, Nat. Commun. 13 (1) (2022).
[4] R. Barrio, A unified rounding error bound for polynomial evaluation, Lect. Notes Pure Appl. 19 (2003).
[5] I. Batatia, et al., MACE: higher order equivariant message passing neural networks for fast and accurate force fields, Adv. Neural Inf. Process. Syst. 35 (2022).
[6] H.E. Bell, Gershgorin's theorem and the zeros of polynomials, Am. Math. Mon. 72 (3) (1965).
[7] J.P. Boyd, Computing zeros on a real interval through Chebyshev expansion and polynomial rootfinding, SIAM J. Numer. Anal. 40 (5) (2002).
[8] A.H. Castro Neto, et al., The electronic properties of graphene, Rev. Mod. Phys. 81 (1) (2009).
[9] C.W. Clenshaw, A note on the summation of Chebyshev series, Math. Comput. 9 (51) (1955).
[10] W.J. Cody, A survey of practical rational and polynomial approximation of functions, SIAM Rev. 12 (3) (1970).
[11] F.W.J. Olver, A.B. Olde Daalhuis, D.W. Lozier, B.I. Schneider, R.F. Boisvert, C.W. Clark, B.R. Miller, B.V. Saunders, H.S. Cohl, M.A. McClain (Eds.), NIST Digital Library of Mathematical Functions, https://dlmf.nist.gov/.
[12] P.O. Dral, M. Barbatti, Molecular excited states through a machine learning lens, Nat. Rev., Chem. 5 (6) (2021).
[13] R. Drautz, Atomic cluster expansion for accurate and transferable interatomic potentials, Phys. Rev. B 99 (2019).
[14] G. Dusson, et al., Atomic cluster expansion: completeness, efficiency and stability, J. Comput. Phys. 454 (2022).
[15] A. Edelman, H. Murakami, Polynomial roots from companion matrix eigenvalues, Math. Comput. 64 (210) (1995).
[16] E.S. Egge, An Introduction to Symmetric Functions and Their Combinatorics, vol. 91, American Mathematical Soc., 2019.
[17] D. Elliott, Error analysis of an algorithm for summing certain finite series, J. Aust. Math. Soc. 8 (2) (1968).
[18] N. Fedik, et al., Extending machine learning beyond interatomic potentials for predicting molecular properties, Nat. Rev., Chem. 6 (9) (2022).
[19] M. Fiedler, A note on companion matrices, Linear Algebra Appl. 372 (2003).
[20] M. Fiedler, Expressing a polynomial as the characteristic polynomial of a symmetric matrix, Linear Algebra Appl. 141 (1990).
[21] L. Fox, I.B. Parker, Chebyshev Polynomials in Numerical Analysis, Oxford University Press, 1968.
[22] G. Frobenius, Theorie der linearen Formen mit ganzen Coefficienten, J. Reine Angew. Math. 86 (1879) (in German).
[23] H.G. Funkhouser, A short account of the history of symmetric functions of roots of equations, Am. Math. Mon. 37 (7) (1930).
[24] W. Gautschi, The condition of orthogonal polynomials, Math. Comput. 26 (120) (1972).
[25] W. Gautschi, The condition of polynomials in power form, Math. Comput. 33 (145) (1979).
[26] S. Gerschgorin, Über Die Abgrenzung der Eigenwerte einer Matrix, Izv. Math. 6 (1931), https://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=im&paperid=5235&option_lang=eng (visited on 04/10/2025).
[27] A. Girard, Invention Nouvelle En L'algebre (reprint) (in French), Ed. by D. Bierens de Haan, 1884.
[28] G.H. Golub, Bounds for eigenvalues of tridiagonal symmetric matrices computed by the LR method, Math. Comput. 16 (80) (1962).

[29] I.J. Good, The colleague matrix, a Chebyshev analogue of the companion matrix, Q. J. Math. 12 (1) (1961).
[30] R.W. Hamming, Introduction to Applied Numerical Analysis, Unabridged republication of the ed. 1989, Dover Publications, Inc., 2012.
[31] M. Hladík, Eigenvalues of symmetric tridiagonal interval matrices revisited, arXiv:1704.03670, 2017.
[32] M. Hladík, D. Daney, E. Tsigaridas, Bounds on real eigenvalues and singular values of interval matrices, SIAM J. Matrix Anal. Appl. 31 (4) (2010).
[33] M. Hladík, D. Daney, E. Tsigaridas, Characterizing and approximating eigenvalue sets of symmetric interval matrices, Comput. Math. Appl. 62 (8) (2011).
[34] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, 2012.
[35] Wolfram Research Inc., Mathematica, Version 13.3, Champaign, IL, 2023.
[36] H. Köuppel, W. Domcke, L.S. Cederbaum, Multimode molecular dynamics beyond the born-Oppenheimer approximation, Adv. Chem. Phys. (1984).
[37] H.J. Kulik, et al., Roadmap on machine learning in electronic structure, Electron. Struct. 4 (2) (2022).
[38] C. Lanczos, Applied Analysis, Dover Publications, 1988.
[39] J.M. Liu, I.T. Lin, Graphene Photonics, Cambridge University Press, 2018.
[40] N. Loehr, Combinatorics, second edition, CRC Press, 2017.
[41] A. Loewy, Begleitmatrizen und Lineare Homogene Differentialausdrücke (in German), Math. Z. 7 (1) (1920).
[42] Username: J.M., Answer to "Small Issue with Chebyshev Derivative Approximation", https://mathematica.stackexchange.com/a/13681, 2012 (visited on 04/09/2025).
[43] C.C. MacDuffee, The Theory of Matrices, Springer, 1933.
[44] D.S. Mackey, The continuing influence of Fiedler's work on companion matrices, Linear Algebra Appl. 439 (4) (2013).
[45] S. Mai, L. González, Molecular photochemistry: recent developments in theory, Angew. Chem., Int. Ed. Engl. 59 (39) (2020).
[46] G. Mayer, Interval Analysis and Automatic Result Verification, De Gruyter, 2017.
[47] R.E. Moore, R.B. Kearfott, M.J. Cloud, Introduction to Interval Analysis, SIAM, 2009.
[48] R.G. Mosier, Root neighborhoods of a polynomial, Math. Comput. 47 (175) (1986).
[49] Y. Nakatsukasa, V. Noferini, On the stability of computing polynomial roots via confederate linearizations, Math. Comput. 85 (301) (2016).
[50] V. Noferini, J. Pérez, Chebyshev rootfinding via computing eigenvalues of colleague matrices: when is it stable?, Math. Comput. 86 (306) (2017).
[51] V. Noferini, L. Robol, R. Vandebril, Structured backward errors in linearizations, Electron. Trans. Numer. Anal. 54 (2021).
[52] D. Opalka, W. Domcke, Interpolation of multi-sheeted multi-dimensional potential-energy surfaces via a linear optimization procedure, J. Chem. Phys. 138 (2013) 22.
[53] B.N. Parlett, The Symmetric Eigenvalue Problem, SIAM, 1998.
[54] A.S. Phani, M.I. Hussein (Eds.), Dynamics of Lattice Materials, John Wiley & Sons, 2017.
[55] F. Plasser, et al., Highly efficient surface hopping dynamics using a linear vibronic coupling model, Phys. Chem. Chem. Phys. 21 (1) (2019).
[56] T.J. Rivlin, Chebyshev Polynomials: From Approximation Theory to Algebra & Number Theory, second edition, Dover Publications, Inc., 2020.
[57] J. Rohn, A Handbook of Results on Interval Linear Problems, Technical report No. V-1163, Institute of Computer Science, Academy of Sciences of the Czech Republic, 2005.
[58] S.M. Sadat, R.Y. Wang, A machine learning based approach for phononic crystal property discovery, J. Appl. Phys. 128 (2020) 2.
[59] H.E. Salzer, A recurrence scheme for converting from one orthogonal expansion into another, Commun. ACM 16 (11) (1973).
[60] G. Schmeisser, A real symmetric tridiagonal matrix with a given characteristic polynomial, Linear Algebra Appl. 193 (1993).
[61] G.W. Semenoff, Condensed-matter simulation of a three-dimensional anomaly, Phys. Rev. Lett. 53 (26) (1984).
[62] K. Serkh, V. Rokhlin, A provably componentwise backward stable $O(n^2)$ QR algorithm for the diagonalization of colleague matrices, arXiv:2102.12186, 2021.
[63] G.A. Sitton, et al., Factoring very-high-degree polynomials, IEEE Signal Process. Mag. 20 (6) (2003).
[64] W. Specht, Die Lage der Nullstellen eines Polynoms, Math. Nachr. 15 (5–6) (1956) (in German).
[65] R.P. Stanley, S. Fomin, Enumerative Combinatorics, Cambridge University Press, 1999.

[66] G. Strang, Introduction to Linear Algebra, sixth edition, SIAM, 2022.
[67] H.C. Thacher Jr., Conversion of a power to a series of Chebyshev polynomials, Commun. ACM 7 (3) (1964).
[68] L.N. Trefethen, Approximation Theory and Approximation Practice, extended edition, SIAM, 2019.
[69] L.N. Trefethen, Six myths of polynomial interpolation and quadrature, Math. Today 4 (2011).
[70] L.N. Trefethen, D. Bau III, Numerical Linear Algebra, vol. 181, SIAM, 2022.
[71] W. Tucker, Validated Numerics: A Short Introduction to Rigorous Computations, Princeton University Press, 2011.
[72] F. Vieta, Opera Mathematica (in Latin), Frans Van Schooten (Ed.) (republished by Georg Olms Verlag, 1970), Elzévir, 1646.
[73] Ė.B. Vinberg, A Course in Algebra, American Mathematical Soc., 2003.
[74] P.R. Wallace, The band theory of graphite, Phys. Rev. 71 (9) (1947).
[75] T.Y. Wang, S.P. Neville, M.S. Schuurman, Machine learning seams of conical intersection: a characteristic polynomial approach, J. Phys. Chem. Lett. 14 (35) (2023).
[76] J. Westermayr, M. Gastegger, P. Marquetand, Combining SchNet and SHARC: the SchNarc machine learning approach for excited-state dynamics, J. Phys. Chem. Lett. 11 (10) (2020).
[77] J. Westermayr, P. Marquetand, Machine learning for electronically excited states of molecules, Chem. Rev. 121 (16) (2020).
[78] J. Westermayr, et al., Deep learning study of tyrosine reveals that roaming can lead to photodamage, Nat. Chem. 14 (8) (2022).
[79] I. Wilkinson, et al., Excited state dynamics in SO_2. I. Bound state relaxation studied by time-resolved photoelectron-photoion coincidence spectroscopy, J. Chem. Phys. 140 (2014) 20.
[80] J.H. Wilkinson, Rounding Errors in Algebraic Processes, SIAM, 2023.
[81] J.H. Wilkinson, The Algebraic Eigenvalue Problem, Oxford University Press, 1988.
[82] J.H. Wilkinson, The calculation of eigenvectors by the method of Lanczos, Comput. J. 1 (3) (1958).
[83] J.H. Wilkinson, The evaluation of the zeros of ill-conditioned polynomials. Part I, Numer. Math. 1 (1959).
[84] J.H. Wilkinson, The evaluation of the zeros of ill-conditioned polynomials. Part II, Numer. Math. 1 (1959).
[85] R.P. Xian, et al., A machine learning route between band mapping and band structure, Nat. Comput. Sci. 3 (1) (2023).