Sparse Polynomial Hermite Interpolation

Erich L. Kaltofen
Dept. of Mathematics, NCSU and Dept. of Computer Science, Duke University
Raleigh, Durham, North Carolina, USA

ABSTRACT

We present Hermite polynomial interpolation algorithms that for a sparse univariate polynomial f with coefficients from a field compute the polynomial from fewer points than the classical algorithms. If the interpolating polynomial f has t terms, our algorithms, require argument/value triples $(\omega^i, f(\omega^i), f'(\omega^i))$ for $i = 0, \ldots, t + \lceil (t+1)/2 \rceil - 1$, where ω is randomly sampled and the probability of a correct output is determined from a degree bound for f. With f' we denote the derivative of f. Our algorithms generalize to multivariate polynomials, higher derivatives and sparsity with respect to Chebyshev polynomial bases. We have algorithms that can correct errors in the points by oversampling at a limited number of good values. If an upper bound $B \geq t$ for the number of terms is given, our algorithms use a randomly selected ω and, with high probability, $\lceil t/2 \rceil + B$ triples, but then never return an incorrect output.

The algorithms are based on Prony's sparse interpolation algorithm. While Prony's algorithm and its variants use fewer values, namely, $2t+1$ and $t+B$ values $f(\omega^i)$, respectively, they need more arguments ω^i. The situation mirrors that in algebraic error correcting codes, where the Reed-Solomon code requires fewer values than the multiplicity code, which is based on Hermite interpolation, but the Reed-Solomon code requires more distinct arguments. Our sparse Hermite interpolation algorithms can interpolate polynomials over finite fields and over the complex numbers, and from floating point data. Our Prony-based approach does not encounter the Birkhoff phenomenon of Hermite interpolation, when a gap in the derivative values causes multiple interpolants. We can interpolate from $t+1$ values of f and $2t-1$ values of f'.

CCS CONCEPTS

• **Mathematics of computing → Interpolation**.

KEYWORDS

model fitting; gradient data; measurement minimization;

ACM Reference Format:
Erich L. Kaltofen. 2022. Sparse Polynomial Hermite Interpolation. In *Proceedings of the 47th ACM Symposium on Symbolic and Algebraic Computation (ISSAC '22), July 4–7, 2022, Lille, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3476446.3535501

ISSAC '22, July 4–7, 2022, Villeneuve-d'Ascq, France.

ACM ISBN 978-1-4503-8688-3/22/07…$15.00
https://doi.org/10.1145/3476446.3535501

1 INTRODUCTION

Let $f(x)$ be a sparse Laurent polynomial in standard basis of powers of the variable over a field K,

$$f(x) = \sum_{j=1}^{t} c_j x^{e_j} \in \mathsf{K}\left[x, \frac{1}{x}\right], \quad \forall j\colon c_j \neq 0, e_j \in \mathbb{Z}, e_1 > e_2 > \cdots > e_t. \tag{1}$$

We allow negative e_j from the start so that our algorithms can be transferred to orthogonal polynomial bases [2, 12]; see Section 5.2. The derivative of $f(x)$ in x is defined $f'(x) = \sum_{j=1}^{t} c_j e_j x^{e_j - 1}$, which is a polynomial of sparsity t or $\leq t-1$, the latter if there are $e_j = 0$, possibly taken modulo the positive characteristic of the coefficient field K. The classical Hermite interpolation algorithm can recover f from $\max(e_1, 0) - \min(0, e_t) + 1 = (\ell_1 + 1) + \cdots + (\ell_n + 1)$ values $a_{i,j} = f^{(j)}(\xi_i)$ for $1 \leq i \leq n$ and $0 \leq j \leq \ell_i$, where $\deg(f) \stackrel{\text{def}}{=} \max(e_1, 0) - \min(0, e_t)$ and $f^{(j)}(x) = \mathrm{d}^j f(x)/\mathrm{d}x^j$ is the j-th derivative of f. Here $\xi_i \in \mathsf{K}$ are distinct arguments. The classical Prony sparse interpolation algorithm can recover f from $2t$ values $\xi_i = \omega^i$ for $\omega \in \mathsf{K}$ and $i = i_0, i_0 + 1, \ldots, i_0 + 2t - 1$, provided the term values ω^{e_j} are distinct for $1 \leq j \leq t$ and e_j can be computed from ω^{e_j}. If e_1, e_t are not known before interpolation, bounds can be used; if the sparsity t is not known, either an upper bound B can be used and the number of values becomes $t + B$, or t can be computed by a Monte-Carlo randomized algorithm from $2t + 1$ values [16].

We shall transfer the Prony method to the Hermite interpolation problem. If there are at most first derivatives, that is $\ell_i \leq 1$, it is Prony's idea that the term locator polynomial

$$\Lambda(z) = z^t + \lambda_{t-1} z^{t-1} + \cdots + \lambda_0 = (z - \omega^{e_1}) \cdots (z - \omega^{e_t}) \tag{2}$$

minimally, if squarefree, linearly generates $a_i \stackrel{\text{def}}{=} f(\omega^{i_0+i})$ for an arbitrary integer starting index i_0 and $i = 0, 1, \ldots, 2t-1$. The polynomial $\Lambda(z)$ also generates, not always minimally, $\bar{a}_i \stackrel{\text{def}}{=} \omega^{i_0+i} f'(\omega^{i_0+i})$ for $i = 0, 1, \ldots, 2t-1$, because $x f'(x)$ has a termset $\subseteq \{x^{e_j}\}_{1 \leq j \leq t}$. Therefore, the vector sequence $(\left[\begin{smallmatrix} a_i \\ \bar{a}_i \end{smallmatrix}\right])_{i \geq 0}$ is minimally generated by the term locator polynomial $\Lambda(z)$, which can be computed by the matrix Berlekamp-Massey algorithm [22] using 1 candidate right generator and two candidate right auxiliary polynomials. One could also use a sequence of 2-dimensional row vectors and compute $\Lambda(z)$ as the highest degree invariant factor of the 2×2 right matrix generator polynomial, with 1 auxiliary right polynomial [22]. Here we pursue the goal adopted from Hermite interpolation with error correction (multiplicity code decoding) [18, 19] and determine the minimum number of values and derivative values required in order to always recover the term locator polynomial $\Lambda(z)$. If for all arguments ω^i one has both a_i and $\bar{a}_i$, experiments quickly show that $t + \lceil t/2 \rceil$ arguments ω^{i_0+i} are required. In general for r with $1 \leq r \leq t-1$, if $f(\omega^{i_0+i})$ is input for $0 \leq i \leq t + r - 1$ and $f'(\omega^{i_0+i})$ is input for $0 \leq i \leq 2t - r - 1$ one can always recover f, for fields K of characteristic 0 or sufficiently large or randomly

selected positive characteristic.

We now list the major differences to Prony sparse polynomial interpolation, for the case that highest order of derivatives is = 1; higher derivatives are discussed later.

1. A total of $3t$ values and values of derivatives is minimally required; the Prony algorithm requires $2t$. We assume for now that the sparsity is input. Such an increase in the required number of values is familiar from multiplicity code decoding, but for error correction the increase is a consequence of the Birkhoff problem in Hermite interpolation, when there are gaps in the inputs of the sequence of values $f(\xi_i), f'(\xi_i), f''(\xi_i), \ldots$ at a single ξ_i (cf. [18, Example 1]). Here there can be gaps: in fact, for $r = 1$ one has $t + 1$ values and $2t - 1$ values at derivatives and recovers f.
2. In contrast to Prony's algorithm, the minimal count of number of values requires randomization in the selection of ω. We have examples where for a given $\omega \in \mathsf{K} = \mathbb{C}$ for which the term locator polynomial $\Lambda(z)$ in (2) is squarefree, $3t$ values are insufficient to recover f; see Example 3.3 below.
3. The number of arguments at which the algorithm computes values and/or values at derivatives can be less than the $2t$ of Prony's algorithm, which is again familiar from the multiplicity code of Hermite interpolation with error correction [18, 19]. If first derivatives (gradients) can be obtained at the arguments, one can interpolate from $t+\lceil t/2\rceil$ distinct points. If higher derivatives can be obtained, one requires even fewer points: $t + \lceil t/\ell\rceil$ arguments ξ_i for ℓ-th order derivatives, and a total number of values of $(\ell + 2)t$; see Section 4.
4. For fields K of characteristic 0, and in certain situations for fields K of characteristic $p > 0$ (see Step 4 below), we can compute the term degrees e_j without taking logarithms of the roots of the term locator polynomial. In particular, we compute the term degrees even when evaluating at powers of a root of unity whose order is below the degree of f, provided that the term values remain distinct. The idea was used before for fast sparse interpolation algorithms of polynomial products [3] and of polynomials given by straight line programs [10, 11] where the asymptotic complexity was optimized, and the number of samples could be increased by a constant factor.

1.1 Relation to Previous Work

Sparse approximation by possibly multivariate functions of data that includes gradient information is a highly investigated subject, from local piecewise spline interpolation to global compressive sparsification by ℓ^1-norm optimization [1]. The Prony algorithm, which was originally designed for sums of exponentials [26] and used for sparse polynomials over finite fields to decode the 1959 BCH digital error correction code, is suitable for floating point data [6, 7]. Our algorithms for sparse polynomial Hermite interpolation compute an exact fit, that is, can be used for scalars from a finite field like the multiplicity error correction code [19, 23]. We note that multiplicity code decoding has already been shown to be suitable for floating point data [14]. As in Hermite interpolation with error correction, our algorithms necessarily require more data than the Prony algorithm, but at fewer points. The reduction in the number of measurement points remains effective for higher derivatives (Section 4), in the multivariate setting (Section 5.1), when interpolation sparse polynomials in orthogonal bases (Section 5.2) and when correcting errors (Section 5.3). A special algorithm interpolates f from $f(1), f'(1), \ldots, f^{(2t-1)}(1)$ [25].

2 SPARSE HERMITE INTERPOLATION

For the t-sparse Laurent polynomial $f(x)$ (1) and all integers i_0 the sequences $(a_i)_{i\geq 0}$ with $a_i = f(\omega^{i_0+i})$ and $(\bar{a}_i)_{i\geq 0}$ with $\bar{a}_i = \omega^{i_0+i} f'(\omega^{i_0+i})$ for $\omega \in \mathsf{K}$ are linearly generated by $\Lambda(z)$ (2). Therefore, we have for all r with $1 \leq r \leq t-1$

$$H_{t,r}\begin{bmatrix}\lambda_0\\ \lambda_1\\ \vdots\\ \lambda_{t-1}\end{bmatrix} = h_{t,r} \text{ for } H_{t,r} \stackrel{\text{def}}{=} \begin{bmatrix} a_0 & a_1 & \ldots & a_{t-1}\\ \vdots & \vdots & & \vdots\\ a_{r-1} & a_r & \ldots & a_{r+t-2}\\ \bar{a}_0 & \bar{a}_1 & \ldots & \bar{a}_{t-1}\\ \vdots & \vdots & & \vdots\\ \bar{a}_{t-r-1} & \bar{a}_{t-r} & \ldots & \bar{a}_{2t-r-2}\end{bmatrix}, h_{t,r} \stackrel{\text{def}}{=} \begin{bmatrix} -a_t\\ \vdots\\ -a_{t+r-1}\\ -\bar{a}_t\\ \vdots\\ -\bar{a}_{2t-r-1}\end{bmatrix}. \quad (3)$$

The condition under which the term locator polynomial Λ is the minimal generator can be characterized for each sequence separately. For the sequence $(a_i)_{i\geq 0}$ a necessary and sufficient condition is that the ω^{e_j} are distinct elements in K (see, for instance, [2, Lemma 4.2]). We will prove in Section 3 that for randomly sampled ω the matrix $H_{t,r}$ is non-singular with high probability, for all r and all fields K of characteristic 0; for fields K of characteristic $p > 0$, the statement remains valid if the prime p is sufficiently large (see Theorem 3.2) or if p can be randomly chosen and f projected modulo p. However, $\det(H_{t,r}) = 0$ is possible for fields K of characteristic 0 even if all ω^{e_j} are distinct: see Example 3.3 below.

Our sparse Hermite interpolation algorithm randomly selects ω, checks the matrix $H_{t,r}$ for non-singularity, and then proceeds similar to Prony's algorithm. Since the term exponents are computed differently, we explicitly state the algorithm. Because the algorithm uses f as a black box, it is useful to flag a false sparsity t or erroneous a_i or $\bar{a}_i$ if that can be diagnosed.

2.1 Algorithm for Explicit Sparsity Input

Input: Let $f(x) = \sum_{j=1}^{t} c_j x^{e_j} \in \mathsf{K}[x, \frac{1}{x}]$ with $t \geq 2$, $c_j \in \mathsf{K}$, $c_j \neq 0$, $e_j \in \mathbb{Z}$ for all j, $e_1 > \cdots > e_t$. The field of scalars K is of characteristic 0 or $p > 0$. The inputs are:

- The number t of terms in f and an r with $1 \leq r \leq t-1$;
- an element $\omega \in \mathsf{K}$, $\omega \neq 0$, $\omega \neq 1$, and
- $i_0 \in \mathbb{Z}$ and values $a_i = f(\omega^{i_0+i})$ for $0 \leq i \leq t+r-1$, $\bar{a}_i = \omega^{i_0+i} f'(\omega^{i_0+i})$ for $0 \leq i \leq 2t-r-1$ such that $\det(H_{t,r}) \neq 0$ where $H_{t,r}$ is defined in (3).

Output:

- Case 1(a): K has characteristic 0 or
  Case 1(b): K has characteristic $p > 0$ and $\omega^\nu \neq 1$ for all $\nu \geq 1$:
  $c_1, \ldots, c_t$ and $e_1, \ldots, e_t$.
- Case 2: K has characteristic $p > 0$ and $\exists\, \theta \geq 2\colon \omega^\theta = 1$:
  $c_1, \ldots, c_t$ and $\tilde{e}_1, \ldots, \tilde{e}_t$ such that for $\tilde{f}(x) \stackrel{\text{def}}{=} \sum_{j=1}^{t} c_j x^{\tilde{e}_j}$ we have $a_i = \tilde{f}(\omega^{i_0+i})$, $0 \leq i \leq t+r-1$ and $\bar{a}_i = \omega^{i_0+i}\tilde{f}'(\omega^{i_0+i})$, $0 \leq i \leq 2t-r-1$.

1. *Solve the $t\times t$ linear system (3) in $\vec{\lambda}$.* See [22, Table 1, Column 2] for fast algorithms. Because $H_{t,r}$ on input is non-singular, $\Lambda(z) = z^t + \sum_{j=0}^{t-1} \lambda_j z^j = \prod_{j=1}^{t}(z - \omega^{e_j})$, and the roots ω^{e_j} are distinct.
2. *Factor $\Lambda(z) = \prod_{j=1}^{t}(z - b_j)$ in $\mathsf{K}[z]$. If $\Lambda(z)$ does not squarefreely factor into linear factors, return "the $a_i, \bar{a}_i$ do not interpolate a t-sparse polynomial."*

3. *Solve the two transposed Vandermonde linear systems*

$$\begin{bmatrix} 1 & \cdots & 1 \\ b_1 & \cdots & b_t \\ \vdots & & \vdots \\ b_1^{t-1} & \cdots & b_t^{t-1} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_t \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{t-1} \end{bmatrix}, \begin{bmatrix} 1 & \cdots & 1 \\ b_1 & \cdots & b_t \\ \vdots & & \vdots \\ b_1^{t-1} & \cdots & b_t^{t-1} \end{bmatrix} \begin{bmatrix} \bar{c}_1 \\ \bar{c}_2 \\ \vdots \\ \bar{c}_t \end{bmatrix} = \begin{bmatrix} \bar{a}_0 \\ \bar{a}_1 \\ \vdots \\ \bar{a}_{t-1} \end{bmatrix}. \quad (4)$$

See [15, Section 5] for a fast algorithm.
If there exists a $c_j = 0$, then return "the $a_i, \bar{a}_i$ do not interpolate a t-sparse polynomial."

4. *Compute the term exponents $\hat{e}_j = \bar{c}_j / c_j$ for $1 \le j \le t$.*

4(a) Case K has characteristic 0:
If there exists an index j with $\hat{e}_j \notin \mathbb{Z}$ or $\omega^{\hat{e}_j} \neq b_j$ return "the $a_i, \bar{a}_i$ do not interpolate a t-sparse polynomial." Else return $c_1, \ldots, c_t$ and $\hat{e}_1, \ldots, \hat{e}_t$, sorted in descending term degrees.

4(b) Case K has characteristic $p > 0$: for proper inputs, we have $\hat{e}_j \equiv e_j \pmod{p}$. Therefore, *if $\hat{e}_j \notin \mathbb{Z}_p \subseteq \mathsf{K}$, then return "the $a_i, \bar{a}_i$ do not interpolate a t-sparse polynomial."*
If $p > |e_j| = \max(e_j, -e_j)$ then e_j can be computed from $\hat{e}_j$.
For $j = 1, \ldots, t$ do verify $\hat{e}_j$ or $\hat{e}_j - p$ by checking $\omega^{\hat{e}_j} = b_j$ or $\omega^{\hat{e}_j - p} = b_j$; if success in the latter, $\hat{e}_j \leftarrow \hat{e}_j - p$. Note that $\omega^{\hat{e}_j} \neq \omega^{\hat{e}_j - p}$ because $\omega^p \neq 1$.
If all checks succeed, return $c_1, \ldots, c_t$ and $\hat{e}_1, \ldots, \hat{e}_t$, sorted in descending term degrees. Note that even if bounds for e_1, e_t are known, the checks verify the $a_i, \bar{a}_i$ by verifying the Prony generator Λ.
At this point, for all $j \in J \subseteq \{1, \ldots, t\}, J \neq \emptyset$, e_j is too large to be directly determined from $e_j \bmod p$. For those j we require an integer logarithm algorithm, as is required in the original Prony algorithm. We have two subcases.

4(b)i. Subcase $\omega^\theta = 1$ for a $\theta \ge 2$: here $\mathbb{Z}_p(\omega)$ is a finite subfield of K, the latter of which could be an infinite function field. Furthermore, θ divides $|\mathbb{Z}_p(\omega)| = p^\nu - 1$ where $\nu \ge 1$.
For all $j \in J$ compute $\tilde{e}_j \in \mathbb{Z}_{\theta p}$ such that $\omega^{\tilde{e}_j} = b_j$ and $\tilde{e}_j \equiv \hat{e}_j \pmod{p}$. One can Chinese remainder the index (discrete logarithm) of b_j with base ω and $\hat{e}_j$ using the relatively prime moduli θ and p. *If $\tilde{e}_j$ exist for all $j \in J$, return $c_1, \ldots, c_t$ and $\tilde{e}_1, \ldots, \tilde{e}_t$, sorted in descending term degrees.*

4(b)ii. Subcase ω has infinite multiplicative order. *For all $j \in J$ compute $\tilde{e}_j \in \mathbb{Z}$ such that $\omega^{\tilde{e}_j} = b_j$ and verify that $\tilde{e}_j \equiv \hat{e}_j \pmod{p}$. If $\tilde{e}_j$ are computed for all $j \in J$, return $c_1, \ldots, c_t$ and $\tilde{e}_1, \ldots, \tilde{e}_t$, sorted in descending term degrees.* For example, if $\mathsf{K} = \mathbb{Z}_p(u)$ is a rational function field, $\tilde{e}_j$ can be determined from the degrees of $b_j(u)$ and $\omega(u)$.

4(b) concluded. At this point, either Step 4(b)i or Step 4(b)ii failed to compute a term degree for a b_j. *Return "the $a_i, \bar{a}_i$ do not interpolate a t-sparse polynomial."*

Algorithm 2.1 is the basic algorithm. If the sparsity t is not known on input, we assume to have a black box for the values of f and f'. In an online-algorithmic way, one computes values

$$A_i \stackrel{\text{def}}{=} \begin{bmatrix} f(\omega^{i_0+i}) \\ \omega^{i_0+i} f'(\omega^{i_0+i}) \end{bmatrix} = \begin{bmatrix} a_i \\ \bar{a}_i \end{bmatrix} \in \mathsf{K}^2 \text{ for } i = 0, 1, \ldots \quad (5)$$

one-at-a-time and terminates in two ways.

1. one inputs an upper bound $D \ge \deg(f) = \max(e_1, 0) - \min(0, e_t) \ge e_1 - e_0$ and computes the sparsity t by the randomized early termination idea in [16]. The bound D is needed to control the probability of correctness.
2. one inputs an upper bound $B \ge t$ and stops when any further discrepancy in the values would push the term locator polynomial beyond degree B, as in the termination criterion of the matrix Berlekamp-Massey algorithm in [22].

We briefly explain both algorithms, assuming that the scalar field K is of characteristic 0 or $p \ge \deg(f) + 1$. For the randomized early termination algorithm in Item 1 we consider the infinite Hankel matrix with 2-dimension vector entries

$$H_\infty = \begin{bmatrix} A_0 & A_1 & A_2 & \ldots \\ A_1 & A_2 & A_3 & \ldots \\ A_2 & A_3 & A_4 & \ldots \\ \vdots & \vdots & \vdots & \end{bmatrix}. \quad (6)$$

As in [16] we can prove that for symbolic $\omega = x$ all $k \times k$ leading principal submatrices are non-singular for $i_0 \neq 0$; see Theorem 3.5 below. Therefore, for a randomly selected $\omega \in S \subseteq \mathsf{K}$, $0 \notin S$ and $1 \notin S$, the first singular leading principal submatrix is at dimension $t+1$, and t is computed with probability $\ge 1 - (\frac{1}{3}t^3 + \frac{2}{3}t)(e_1 - e_t)/|S| \ge 1 - (\frac{1}{3}t^3 + \frac{2}{3}t)D/|S|$; see Corollary 3.2. One computes $t + \lceil (t+1)/2 \rceil$ values of f and $t + \lfloor (t+1)/2 \rfloor$ values of f'.

If $B \ge t$ is input, one can stop, with the assumption that $H_{t,\lceil t/2 \rceil}$ is non-singular, which for random ω is achieved with high probability,[1] after computing the values a_i for $i = 0, \ldots, \lceil t/2 \rceil + B - 1$ and $\bar{a}_i$ for $i = 0, \ldots, \lfloor t/2 \rfloor + B - 1$, $2B + t$ values in total. We give the argument when $t \equiv 0 \pmod 2$ without appealing to the matrix Berlekamp-Massey stopping criterion [22]. For the matrix

$$\begin{bmatrix} A_0 & A_1 & \ldots & A_{t-1} & A_t \\ \vdots & \vdots & & \vdots & \vdots \\ A_{B-t/2-1} & A_{B-t/2} & \ldots & A_{B+t/2-2} & A_{B+t/2-1} \end{bmatrix} \in \mathsf{K}^{(2B-t)\times(t+1)} \quad (7)$$

the algorithm generates the $(t+1)$'st column by a linear combination of the first t columns, which corresponds to the term locator polynomial $\Lambda(z) = z^t + \lambda_{t-1} z^{t-1} + \cdots + \lambda_0$. The column relation is determined by the first t rows and verified by the next $2B - 2t$ rows. We suppose now that the column relation fails in a subsequent row $\begin{bmatrix} A_L & A_{L+1} & \ldots & A_{L+t-1} & A_{L+t} \end{bmatrix}$ where $L \ge B - t/2$. We only consider a discrepancy in the first row, that is,

$$\delta \stackrel{\text{def}}{=} \lambda_0 a_L + \cdots + \lambda_{t-1} a_{L+t-1} + a_{L+t} \neq 0. \quad (8)$$

As in [21, Proof of Lemma 2], we have

$$\begin{bmatrix} A_0 & \ldots & A_t & \ldots & A_{M-1} \\ A_1 & \ldots & A_{t+1} & \ldots & A_M \\ \vdots & & \vdots & & \vdots \\ A_{t/2-1} & \ldots & A_{3t/2-1} & \ldots & A_{L+t-1} \\ a_{t/2} & \ldots & a_{3t/2} & \ldots & a_{L+t} \\ a_{t/2+1} & \ldots & a_{3t/2+1} & \ldots & a_{L+t+1} \\ \vdots & & \vdots & & \vdots \\ a_L & \ldots & a_{L+t} & \ldots & a_{L+M-1} \end{bmatrix} \begin{bmatrix} 1 & 0 & \ldots & 0 & \lambda_0 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 & \lambda_1 & \lambda_0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 & \lambda_{t-1} & \lambda_{t-2} & \ldots & \\ 0 & 0 & \ldots & 0 & 1 & \lambda_{t-1} & \ldots & \\ & & & & & & & \\ 0 & 0 & \ldots & 0 & 0 & 1 & \ddots & \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 1 \end{bmatrix}$$

$$= \begin{bmatrix} PH_{t,t/2} & 0 & 0 & 0 \\ * & 0 & 0 & \delta \\ & \vdots & \iddots & \\ * & \delta & * & \end{bmatrix}, \quad \begin{array}{l} \text{where } M \stackrel{\text{def}}{=} t + (L - \frac{t}{2} + 1) = L + \frac{t}{2} + 1, \\ P \text{ a (row-)permutation matrix;} \end{array} \quad (9)$$

note that the matrices in (9) are of dimension $M \times M$, with $M \ge B+1$. By the assumption that $\det(H_{t,t/2}) \neq 0$ and $\delta \neq 0$ the first matrix factor in (9) is a non-singular matrix, because the upper triangular second matrix factor and the right-side block lower triangular matrix are both non-singular. Therefore, the last column of the first factor in (9) is linearly independent of the previous columns and

[1] In Example 3.3 a deterministic version using a high order root of unity for ω is discussed.

cannot be linearly generated by the previous columns, implying that the degree of the generator would have to be $\geq M \geq B+1$. The cases where $t \equiv 1 \pmod 2$ and/or when the discrepancy is at $\bar{a}_{L+t}$ are done similarly.

We briefly consider when $H_{t,\lceil t/2\rceil}$ is singular, that is, the randomly selected ω is "unlucky," for instance, an ω in Example 3.3 or an ω with $\omega^{e_\mu} = \omega^{e_\nu}$ for $\mu \neq \nu$. For a $k \leq t$ one has computed a linear combination λ of the first k columns that gives the $(k+1)$'st column at the first $2B-k$ (k even) or $2B-k-1$ (k odd) rows of H_∞ (6), using $2B+k$ values $a_i, \bar{a}_i$. If the top left $k\times k$ matrix is singular, one has verified that ω as unlucky, but one still may recover f if $B > t$. For the column relation λ, Algorithm 2.1, Steps 2–4 are executed. If the algorithm fails to compute an interpolant, one has verified $k < t$. An example is $B = t$ and ω as in Example 3.3: $k \leq t-1$ and the $(k+1)$'s column in $H_{t,\lceil t/2\rceil}$ is linearly dependent on the first k columns, but the $2B+k \leq 3t-1$ values do not determine the term locator polynomial. However, Algorithm 2.1 may compute a k-sparse polynomial $h(x)$ that fits the $a_i, \bar{a}_i$ for $i \leq \lceil k/2\rceil + B$, for example, if $\omega^{e_\mu} = \omega^{e_\nu} = \omega^{e_\kappa}$ for distinct e_μ, e_ν, e_κ (cf. [2, Lemma 4.2]) and $f(x) = c_\mu x^{e_\mu} + c_\nu x^{e_\nu} + c_\kappa x^{e_\kappa}$, $h(x) = (c_\mu + c_\nu + c_\kappa)x^{e_\mu}$ with $e_\mu c_{e_\mu} + e_\nu c_{e_\nu} + e_\kappa c_{e_\kappa} = e_\mu(c_{e_\mu} + c_{e_\nu} + c_{e_\kappa}) \neq 0$. The computed interpolant $h(x)$ fits the $a_i = f(\omega^{i_0+i}) = h(\omega^{i_0+i})$, $\bar{a}_i = \omega^{i_0+i} f'(\omega^{i_0+i}) = \omega^{i_0+i} h'(\omega^{i_0+i})$ for all $i > \lceil k/2\rceil + B$, because if there were a discrepancy with the values of f, f', the next linear generator would have degree $> B$ by a matrix factorization similar to (9) after permuting rows so that the top left $k\times k$ matrix is non-singular, and could not be the term locator polynomial for f. Therefore, unlike the early termination strategy of Item 1 (cf. [12, example after the proof of Theorem 4.3]), from a bound $B \geq t$ no interpolant is returned that does not fit the infinite sequence of evaluations of f.

3 PROBABILISTIC ANALYSIS

We prove that for random ω the matrix $H_{t,r}$ in (3) is non-singular with high probability. The exact statement is given in Theorem 3.2 below. The method follows that of the early termination proof for the Prony algorithm in [16]. We shall first prove that for the value $\omega = x$ and evaluations $\alpha_i(x) = f(x^{i_0+i})$ and $\bar{\alpha}_i(x) = x^{i_0+i} \times \left(f'(x)\big|_{x=x^{i_0+i}}\right)$ for $i = 0, 1, \ldots$ the corresponding symbolic matrix $\mathcal{H}_{t,r}(x) \in \mathsf{K}[x, \frac{1}{x}]$ is non-singular, at least for a field K of characteristic 0. We have the following matrix factorization:

$$\mathcal{H}_{t,r} \stackrel{\text{def}}{=} \begin{bmatrix} \alpha_0 & \alpha_1 & \ldots & \alpha_{t-1} \\ \vdots & \vdots & & \vdots \\ \alpha_{r-1} & \alpha_r & \ldots & \alpha_{r+t-2} \\ \bar{\alpha}_0 & \bar{\alpha}_1 & \ldots & \bar{\alpha}_{t-1} \\ \vdots & \vdots & & \vdots \\ \bar{\alpha}_{t-r-1} & \bar{\alpha}_{t-r} & \ldots & \bar{\alpha}_{2t-r-2} \end{bmatrix} = W_{t,r} \begin{bmatrix} c_1\beta_1^{i_0} & 0 & \ldots & 0 \\ 0 & c_2\beta_2^{i_0} & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & c_t\beta_t^{i_0} \end{bmatrix} \begin{bmatrix} 1 & \beta_1 & \ldots & \beta_1^{t-1} \\ 1 & \beta_2 & \ldots & \beta_2^{t-1} \\ \vdots & \vdots & & \vdots \\ 1 & \beta_t & \ldots & \beta_t^{t-1} \end{bmatrix}, \quad (10)$$

where $W_{t,r}$ is defined in (11) below for $k = t$ and $\beta_j = x^{e_j}$ are the Laurent terms in f. The last 2 matrix factors in (10) are non-singular because $\beta_\mu = x^{e_\mu} \neq \beta_\nu = x^{e_\nu}$ for $1 \leq \mu < \nu \leq t$ and the last factor is a transposed Vandermonde matrix. We prove that the first factor, $W_{t,r}$ is non-singular by virtue that the highest degree term has a non-zero coefficient. The proof is by induction on the dimension.

Theorem 3.1. *Let $k \geq 1$, $0 \leq r \leq k$ and let*

$$W_{k,r} \stackrel{\text{def}}{=} \left.\begin{bmatrix} 1 & 1 & \ldots & 1 \\ \beta_1 & \beta_2 & \ldots & \beta_k \\ \vdots & \vdots & & \vdots \\ \beta_1^{r-1} & \beta_2^{r-1} & \ldots & \beta_k^{r-1} \\ e_1 & e_2 & \ldots & e_k \\ e_1\beta_1 & e_2\beta_2 & \ldots & e_k\beta_k \\ \vdots & \vdots & & \vdots \\ e_1\beta_1^{k-r-1} & e_2\beta_2^{k-r-1} & \ldots & e_k\beta_k^{k-r-1} \end{bmatrix}\right\} \begin{matrix} r \text{ rows} \\ \\ \\ \\ \\ \\ \\ k-r \text{ rows} \end{matrix} \in \mathbb{Z}\left[x, \frac{1}{x}\right]^{k\times k}, \quad (11)$$

where $\beta_j = x^{e_j}$ are Laurent terms with $e_j \in \mathbb{Z}$ for all $1 \leq j \leq k$ with $e_1 > e_2 > \cdots > e_k$ and $e_j \neq 0$ for $r < k/2$ and $1 \leq j \leq k-2r$. Then the leading monomial of $\det(W_{k,r})$ in the variable x is:

$$\sigma(k,r)\Big(\prod_{\nu=1}^{k-2r} e_\nu\Big)\Big(\prod_{\mu=0}^{\min(r,k-r)-1} (e_{k-2\mu-1} - e_{k-2\mu})\Big)\prod_{j=1}^{k-1}\beta_j^{\eta(k,r,j)} \quad (12)$$

where the first product in (12) is $= 1$ for $k \leq 2r$ and the second product in (12) is $= 1$ for $r = 0$ or $r = k$, and where the sign is

$$\sigma(k,r) = \begin{cases} (-1)^{\lfloor k/2\rfloor + \lfloor r/2\rfloor} & \text{for } 0 \leq r < \frac{k}{2}, \\ (-1)^{k-\lceil r/2\rceil} & \text{for } \frac{k}{2} < r \leq k, \\ (-1)^{\lceil k/4\rceil} = (-1)^{k/2+\lfloor k/4\rfloor} = & \\ \quad (-1)^{k-\lceil k/4\rceil} & \text{for } k \equiv 0 \pmod 2 \text{ and } r = \frac{k}{2}, \end{cases} \quad (13)$$

and where the exponents are

$$\eta(k,r,j) = \max\Big(\Big\lceil\frac{k-j-1}{2}\Big\rceil, \max(r, k-r) - j\Big). \quad (14)$$

We have $\eta(k,r,j) \geq \eta(k,r,j+1)$ for all j with $1 \leq j \leq k-1$.

Specifically, for $r = \lceil k/2\rceil$ the leading monomial is:

$$(-1)^{\lceil k/4\rceil}\ (e_1 - e_2)(e_3 - e_4)\cdots(e_{k-1} - e_k) \times (\beta_1\beta_2)^{k/2-1}(\beta_3\beta_4)^{k/2-2}\cdots(\beta_{k-3}\beta_{k-2}) \text{ if } k \text{ is even}, \quad (15)$$

$$(-1)^{\lceil k/4\rceil+1}\ (e_2 - e_3)(e_4 - e_5)\cdots(e_{k-1} - e_k)\beta_1^{(k+1)/2-1} \times (\beta_2\beta_3)^{(k+1)/2-2}(\beta_4\beta_5)^{(k+1)/2-3}\cdots(\beta_{k-3}\beta_{k-2}) \text{ if } k \text{ is odd}. \quad (16)$$

For $k \equiv 1 \pmod 2$ and $r = \lfloor k/2\rfloor$ the leading monomial is:

$$(-1)^{\lceil (k-1)/4\rceil} e_1(e_2 - e_3)(e_4 - e_5)\cdots(e_{k-1} - e_k)\ \beta_1^{(k-1)/2}(\beta_2\beta_3)^{(k-1)/2-1}\cdots(\beta_{k-3}\beta_{k-2}). \quad (17)$$

Proof. The formulas are proven by induction with minor expansion along the first column of $W_{k,r}$ in (11), as if arising by themselves.[2] See Section 7. □

The condition that $e_j \neq 0$ for $r < k/2$ and $j \leq k-2r$ in Theorem 3.1 is required so that the factor $\prod_{\nu=1}^{\max(0,k-2r)} e_\nu \neq 0$ in (12). If there is an index $j^* < k$ with $e_{j^*} = 0$, which over $\mathbb{Z}$ implies $e_k < 0$, that is, there are terms with negative degree, the corresponding j^*-th column in $W_{k,r}$ in (11) contains zeros, and the leading monomial changes. Note that for $e_k = 0$ the leading monomial (12) is valid for $r \geq 1$ because the factor e_k only appears for $r = 0$, in which case $\det(W_{k,0}) = 0$. The case in which an $e_{j^*} = 0$ for $j^* < k$ can be reduced to Theorem 3.1 as follows:

[2] See the proof of Theorem 4.1 for a full explanation.

Corollary 3.1. *Let $k \geq 1$ and*

$$W_{k,r}^{[j^*]} \stackrel{\text{def}}{=} \begin{bmatrix} 1 & \cdots & 1 & 1 & 1 & \cdots & 1 \\ \beta_1 & \cdots & \beta_{j^*-1} & 1 & \beta_{j^*+1} & \cdots & \beta_k \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \beta_1^{r-1} & \cdots & \beta_{j^*-1}^{r-1} & 1 & \beta_{j^*+1}^{r-1} & \cdots & \beta_k^{r-1} \\ e_1 & \cdots & e_{j^*-1} & 0 & e_{j^*+1} & \cdots & e_k \\ e_1\beta_1 & \cdots & e_{j^*-1}\beta_{j^*-1} & 0 & e_{j^*+1}\beta_{j^*+1} & \cdots & e_k\beta_k \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ e_1\beta_1^{k-r-1} & \cdots & e_{j^*-1}\beta_{j^*-1}^{k-r-1} & 0 & e_{j^*+1}\beta_{j^*+1}^{k-r-1} & \cdots & e_k\beta_k^{k-r-1} \end{bmatrix}$$

$$\in \mathbb{Z}\big[x, \frac{1}{x}\big]^{k\times k}, 1 \leq r < \frac{k}{2}, 1 \leq j^* \leq k-2r, \quad (18)$$

where $\beta_j = x^{e_j}$ are Laurent terms with $e_j \in \mathbb{Z}$ for all $1 \leq j \leq k$ with $e_1 > \cdots > e_{j^*-1} > e_{j^*} = 0 > e_{j^*+1} > \cdots > e_k$; note that $\beta_{j^*} = 1$. Let $\tau(k, r, j^*, j)$ be the permutation on $\{1, \ldots, k\}$ with

$$\tau(k, r, j^*, j) \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} j & \text{for } j \leq j^*-1 \text{ or } j \geq k-2r+2, \\ j^* & \text{for } j = k-2r+1, \\ j+1 & \text{for } j^* \leq j \leq k-2r. \end{array} \right\} \quad (19)$$

Then the leading monomial of $\det(W_{k,r}^{[j^*]})$ in the variable x is:

$$(-1)^{k-2r-j^*+1}\sigma(k,r)\Big(\prod_{\nu=1}^{k-2r} e_{\tau(k,r,j^*,\nu)}\Big) \times \Big(\prod_{\mu=0}^{\min(r,k-r)-1} (e_{\tau(k,r,j^*,k-2\mu-1)} - e_{\tau(k,r,j^*,k-2\mu)})\Big) \prod_{j=1}^{k-1} \beta_{\tau(k,r,j^*,j)}^{\eta(k,r,j)}, \quad (20)$$

where β_{j^*}=1 and e_{j^*}=0, and where σ is defined in (13) and η in (14).

Proof. All terms in $\det(W_{k,r})$ when keeping e_j and β_j as variables factor like the lead term in (12) using a different variable ordering for the β_j. Substituting the variable $\beta_j = x^{e_{\tau(k,r,j^*,j)}}$ and the variable e_j with the numeric value $e_{\tau(k,r,j^*,j)}$ all terms corresponding to a variable orderings for the β_j which place β_{j^*} before the $(k-2r+1)$-st position and which would have on evaluation a higher degree in x have a zero cofficient because of the factor $\prod_{\nu=1}^{k-2r} e_\nu$ in (12). □

Note that $W_{t,r}$ or $W_{t,r}^{[j^*]}$ can be non-singular even if there is a pair $e_\mu = e_\nu$ with $\mu \neq \nu$, for instance in (16) if $e_1 = e_2$. The matrix $\mathcal{H}_{t,r}$ is then singular by virtue that the Vandermonde factor in (10) is singular.

Theorem 3.2. *Let f, t, r, i_0 be as in the input specifications of Algorithm 2.1. Let K be a field of characteristic 0 or $p \geq \deg(f)+1 = \max(e_1, 0) - \min(e_t, 0) + 1$. Suppose the element ω is randomly and uniformly selected from a finite set $S \subseteq \mathsf{K}$ with $0 \notin S$ and $1 \notin S$ of cardinality $|S|$. Then the probability that $H_{t,r}$ in (3) and Step 1 is non-singular is $\geq 1 - (t-1)^2(e_1 - e_t)/|S|$.*

Proof. By the matrix factorization (10) and by Theorem 3.1 and Corollary 3.1 the $\det(\mathcal{H}_{t,r}) \neq 0$, both in characteristic 0 and $p \geq \deg(f)+1$, the latter because the coefficients of the leading monomials of $W_{t,r}$ (12) and $W_{t,r}^{[j^*]}$ (20) are non-zero modulo p. It remains to estimate the degree in x of the leading and trailing monomial of both $\det(W_{t,r})$ and the determinant of the transposed Vandermonde factor (10). Note that the diagonal factor in (10) is non-singular for all $\omega \in S$. The degree of the leading monomial in $\det(W_{t,r})$ is by (12,20)

$$\leq \sum_{j=1}^{t-2} e_j\eta(t,r,j) \leq \sum_{j=1}^{t-2}(e_1 - j + 1)(t-j-1)$$

$$= \frac{1}{6}(t-1)(t-2)(3e_1 - t + 3). \quad (21)$$

By reversing the monomial order $\beta_1 \prec \ldots \prec \beta_t$ we have $(-1)\times$ the degree of trailing monomial in $\det(W_{t,r})$ to be bounded from below by $-(t-1)(t-2)(3e_t - t + 3)/6$. For the determinant of the right transposed Vandermonde factor the corresponding bounds are $\pm t(t-1)(3e_j - t + 2)/6$ for $j = 1, t$. Summing the two differences of the upper and lower bounds yields a degree bound $(t-1)^2(e_1 - e_t)$. Sharper bounds can be derived for a given r from the first estimate in (21). □

Example 3.3. When computing the determinant of $W_{t,r}$ as a polynomial in x one obtains values $\omega \in \mathsf{K}$ for which $\omega^{e_\mu} \neq \omega^{e_\nu}$ for all $1 \leq \mu < \nu \leq t$ but for which $\det(H_{t,r}) = 0$ (where $H_{t,r}$ is defined in (3)). For example for $t = 3, r = 1$ and $e_2 = e_1 - 1, e_3 = e_1 - 2$ we have

$$\det(W_{3,1}) = \det\left(\begin{bmatrix} 1 & 1 & 1 \\ e_1 & e_1-1 & e_1-2 \\ e_1x^{e_1} & (e_1-1)x^{e_1-1} & (e_1-2)x^{e_1-2} \end{bmatrix}\right)$$

$$= -x^{e_1-2}(x-1)(e_1x - e_1 + 2). \quad (22)$$

Therefore, for $\omega = 1 - \frac{2}{e_1}$ we have $\det(H_{t,r}) = 0$ and $f_1(x) = c_1x^{e_1} + c_2x^{e_1-1} + c_3x^{e_1-2}$ cannot be interpolated from $a_i = f_1(\,(1-\frac{2}{e_1})^i\,)$ for $i = i_0, \ldots, i_0+3$ and $\bar{a}_i = (1-\frac{2}{e_1})^i f_1'(\,(1-\frac{2}{e_1})^i\,)$ for $i = i_0, \ldots, i_0+4$. If $\omega \notin \{0, 1, -1, 1-\frac{2}{e_1}\}$, Algorithm 2.1 interpolates f_1 from 9 values. Here $\omega \neq -1$ because one needs $\omega^{e_1} \neq \omega^{e_3} = \omega^{e_1-2}$.

We give a second example with an equal number of a_i and $\bar{a}_i$. Let $t = 4$, $r = 2$, $e_2 = e_1 - 1$, $e_3 = e_1 - 5$, $e_4 = e_1 - 6$. Then $\det(W_{4,2}) = -x^{2e_1-11}(x^2+3x+1)(x^4+x^3+6x^2+x+1)(x-1)^4$. For $\omega = \frac{1}{2}(3-\sqrt{5}) \approx -0.38197$ the vector sequence $(\left[\begin{smallmatrix} a_i \\ \bar{a}_i \end{smallmatrix}\right])_{i=i_0,\ldots,i_0+5}$ of 6 vectors is insufficient to interpolate $f_2(x) = c_1x^{e_1} + c_2x^{e_1-1} + c_3x^{e_1-5} + c_4x^{e_1-6}$. Note that for $\omega = 2$, Algorithm 2.1 interpolates f_2 from 12 values, which is fewer values than the classical Hermite algorithm for $e_1 \geq 12$.

We do not know if a root of unity $\omega \in \mathbb{C}$ for which $\omega^{e_\mu} \neq \omega^{e_\nu}$ for all $1 \leq \mu < \nu \leq t$ always yields $\det(H_{t,r}) \neq 0$. A root of unity $\omega \in \mathbb{C}$ of prime order $\geq (t-1)^2(e_1-e_t)+2$ always has $\det(H_{t,r}) \neq 0$ by Theorem 3.2. □

Remark 3.4. Corollary 3.1 considers $W_{k,r}^{[j^*]}$ (18) as a polynomial matrix with integer coefficients. When taking its determinant modulo the characteristic $p \geq 2$ of the coefficient field K of f in (1), the coefficient of the term (20) can map to zero. However, a lower degree term will survive if there exists a permutation τ on $\{1, \ldots, k\}$ such that $\big(\prod_{\nu=1}^{k-2r} e_{\tau(\nu)}\big)\big(\prod_{\mu=0}^{\min(r,k-r)-1}(e_{\tau(k-2\mu-1)} - e_{\tau(k-2\mu)})\big) \not\equiv 0 \pmod p$. Note that it is possible that $\det(W_{k,r}^{[j^*]}) \equiv 0 \pmod p$ for distinct term values $\omega^{e_\mu} \neq \omega^{e_\nu}$ for all $1 \leq \mu < \nu \leq t$, for instance, if $e_1 \equiv e_2 \equiv 0 \pmod p$ and $r = 1$. □

The next theorem is used for estimating the probability of success when computing the sparsity t by the randomized early termination strategy.

Theorem 3.5. *Let K be a field of characteristic 0 or $p \geq \deg(f) + 1 = \max(e_1, 0) - \min(e_t, 0) + 1$. Then $\mathcal{H}_{k,\lceil k/2\rceil} \in \mathsf{K}[x]^{k\times k}$ (10) is*

non-singular for $i_0 \neq 0$ and for all $k = 1, 2, \ldots, t$.

Proof. The proof follows directly from [16]. The highest order term in $\mathcal{H}_{k,\lceil k/2\rceil}$ is for even k, for example, by the Cauchy-Binet argument of [16, Proof of Theorem 4] and by (15)

$$\beta_1^{i_0+k-1+k/2-1}\beta_2^{i_0+k-2+k/2-1}\beta_3^{i_0+k-3+k/2-2}\beta_4^{i_0+k-4+k/2-2} \cdots \beta_{k-3}^{i_0+3+1}\beta_{k-2}^{i_0+2+1}\beta_{k-1}^{i_0+1}\beta_k^{i_0}. \quad (23)$$

We note that in [8, Theorem 4.1] the condition $i_0 \neq 0$ is removed for the Prony algorithm for fields K of characteristic $\neq 2$, but we do not know if the argument can be generalized to $\mathcal{H}_{k,\lceil k/2\rceil}$ for $k \geq 2$. □

We now give a very rough estimate on the probability that for a random ω the matrices $H_{k,\lceil k/2\rceil}$ are non-singular for all $1 \leq k \leq t$.

Corollary 3.2. *Let f, t, r, i_0 be as in the input specifications of Algorithm 2.1. Let* K *be a field of characteristic* 0 *or $p \geq \deg(f)+1 = \max(e_1, 0) - \min(e_t, 0) + 1$. Suppose the element ω is randomly and uniformly selected from a finite set $S \subseteq$ K with $0 \notin S$ and $1 \notin S$ of cardinality $|S|$. Then the probability that $H_{k,\lceil k/2\rceil}$ in (3) is non-singular for all k with $1 \leq k \leq t$ is $\geq 1 - \left(\frac{1}{3}t^3 + \frac{2}{3}t\right)(e_1 - e_t)/|S|$.*

Proof. Each $\det(H_{k,\lceil k/2\rceil})$ has degree $\leq (k-1)^2(e_1 - e_t) + k(e_1 - e_t)$ by the estimates in the proof of Theorem 3.2 and (10), after removing the factor $\beta_1^{i_0-1}\cdots\beta_k^{i_0-1}$, (cf. (23)). The sum for all $k = 1, \ldots, t$ is $\frac{1}{6}t(t-1)(2t-1) + \frac{1}{2}t(t+1) = \frac{1}{3}t^3 + \frac{2}{3}t$. Note that the bound is not sharp. □

4 HIGHER DERIVATIVES

The algorithm generalizes to higher derivatives. We only discuss the case when the highest derivative values are second derivatives, and the derivative values are evenly distributed. We suppose that second derivatives values $\bar{a}_i = \omega^{2(i_0+i)} f''(\omega^{i_0+i})$ for $i = 0, 1, \ldots$ are also available for interpolation. The linear system (3) in Step 1 now is

$$H_{t,\lfloor t/3\rfloor+\rho_0,\lfloor t/3\rfloor+\rho_1}\vec{\lambda} = h_{t,\lfloor t/3\rfloor+\rho_0,\lfloor t/3\rfloor+\rho_1} \quad (24)$$

with $\rho_0 = \rho_1 = 0$ for $t \equiv 0 \pmod 3$, $\rho_0 = 1, \rho_1 = 0$ for $t \equiv 1 \pmod 3$, $\rho_0 = 1, \rho_1 = 1$ for $t \equiv 2 \pmod 3$, and where the augmented coefficient matrix $[H_{t,r_0,r_1} \mid h_{t,r_0,r_1}]$ of (24) is (25):

$$\left[\begin{array}{cccc|c} a_0 & a_1 & \ldots & a_{t-1} & -a_t \\ \vdots & \vdots & & \vdots & \vdots \\ a_{r_0-1} & a_{r_0} & \ldots & a_{r_0+t-2} & -a_{r_0+t-1} \\ \bar{a}_0 & \bar{a}_1 & \ldots & \bar{a}_{t-1} & -\bar{a}_t \\ \vdots & \vdots & & \vdots & \vdots \\ \bar{a}_{r_1-1} & \bar{a}_{r_1} & \ldots & \bar{a}_{r_1+t-2} & -\bar{a}_{r_1+t-1} \\ \bar{\bar{a}}_0 & \bar{\bar{a}}_1 & \ldots & \bar{\bar{a}}_{t-1} & -\bar{\bar{a}}_t \\ \vdots & \vdots & & \vdots & \vdots \\ \bar{\bar{a}}_{t-r_0-r_1-1} & \bar{\bar{a}}_{t-r_0-r_1} & \ldots & \bar{\bar{a}}_{2t-r_0-r_1-2} & -\bar{\bar{a}}_{2t-r_0-r_1-1} \end{array}\right]. \quad (25)$$

In total, we need $4t$ values at $t + \lceil t/3\rceil$ arguments $\xi_i = \omega^{i_0+i}$. The coefficient matrix H_{t,r_0,r_1} in (25) is with high probability non-singular by Theorem 4.1 below, which corresponds to (15,16) in Theorem 3.1 above.

Theorem 4.1. *Let $k \geq 1$ and let where $\beta_j = x^{e_j}$ are Laurent terms with $e_j \in \mathbb{Z}$ for all $1 \leq j \leq k$ with $e_1 > e_2 > \cdots > e_k$. Then the leading monomial in the variable x of $\det(W_{k,\lfloor k/3\rfloor+\kappa_0,\lfloor k/3\rfloor+\kappa_1})$, where $W_{k,r_0,r_1} \in \mathbb{Z}\left[x, \frac{1}{x}\right]^{k\times k}$, $0 \leq r_0 \leq k$, $0 \leq r_1 \leq k - r_0$ is (26),*

$$\left[\begin{array}{cccc} 1 & 1 & \ldots & 1 \\ \beta_1 & \beta_2 & \ldots & \beta_k \\ \vdots & \vdots & & \vdots \\ \beta_1^{r_0-1} & \beta_2^{r_0-1} & \ldots & \beta_k^{r_0-1} \\ e_1 & e_2 & \ldots & e_k \\ e_1\beta_1 & e_2\beta_2 & \ldots & e_k\beta_k \\ \vdots & \vdots & & \vdots \\ e_1\beta_1^{r_1-1} & e_2\beta_2^{r_1-1} & \ldots & e_k\beta_k^{r_1-1} \\ e_1(e_1-1) & e_2(e_2-1) & \ldots & e_k(e_k-1) \\ e_1(e_1-1)\beta_1 & e_2(e_2-1)\beta_2 & \ldots & e_k(e_k-1)\beta_k \\ \vdots & \vdots & & \vdots \\ e_1(e_1-1)\beta_1^{k-r_0-r_1-1} & e_2(e_2-1)\beta_2^{k-r_0-r_1-1} & \ldots & e_k(e_k-1)\beta_k^{k-r_0-r_1-1} \end{array}\right], \quad (26)$$

is the following if $k \equiv 0 \pmod 3$ and $\kappa_0 = \kappa_1 = 0$:

$$(-1)^{k/3}\left(\prod_{\mu\equiv 1 \pmod 3}^{k-2}(e_\mu - e_{\mu+1})(e_\mu - e_{\mu+2})\right)\left(\prod_{\nu\equiv 2 \pmod 3}^{k-1}(e_\nu - e_{\nu+1})\right)\times \prod_{j=1}^{k/3-1}(\beta_{3j-2}\beta_{3j-1}\beta_{3j})^{k/3-j}; \quad (27)$$

if $k \equiv 1 \pmod 3$ and $\kappa_0 = 1$ and $\kappa_1 = 0$:

$$(+1)\left(\prod_{\mu\equiv 2 \pmod 3}^{k-2}(e_\mu - e_{\mu+1})(e_\mu - e_{\mu+2})\right)\left(\prod_{\nu\equiv 0 \pmod 3}^{k-1}(e_\nu - e_{\nu+1})\right)\times \beta_1^{(k-1)/3}\prod_{j=1}^{(k-1)/3-1}(\beta_{3j-1}\beta_{3j}\beta_{3j+1})^{(k-1)/3-j}; \quad (28)$$

if $k \equiv 2 \pmod 3$ and $\kappa_0 = 1$ and $\kappa_1 = 1$:

$$(-1)\left(\prod_{\mu\equiv 0 \pmod 3}^{k-2}(e_\mu - e_{\mu+1})(e_\mu - e_{\mu+2})\right)\left(\prod_{\nu\equiv 1 \pmod 3}^{k-1}(e_\nu - e_{\nu+1})\right)\times (\beta_1\beta_2)^{(k-2)/3}\prod_{j=1}^{(k-2)/3-1}(\beta_{3j}\beta_{3j+1}\beta_{3j+2})^{(k-2)/3-j}. \quad (29)$$

Proof. As for Theorem 3.1, the proof is by induction on k. We write $\mathrm{FLM}(k, r_0, r_1; e_1, \ldots, e_k)$ for the formal leading monomial of $\det(W_{k,r_0,r_1})$, when the e_j coefficients are symbols. The terms β_j shall retain their numeric degrees. For $k \equiv 0 \pmod 3$ we obtain as the formal leading monomial:

$$\mathrm{FLM}(k, k/3, k/3; e_1, \ldots, e_k) = \det\left(\left[\begin{array}{ccc} \beta_1^{k/3-1} & \beta_2^{k/3-1} & \beta_3^{k/3-1} \\ e_1\beta_1^{k/3-1} & e_2\beta_2^{k/3-1} & e_3\beta_3^{k/3-1} \\ e_1(e_1-1)\beta_1^{k/3-1} & e_2(e_2-1)\beta_2^{k/3-1} & e_3(e_3-1)\beta_3^{k/3-1} \end{array}\right]\right)\times \mathrm{FLM}(k-3, k/3-1, k/3-1; e_4, \ldots, e_k). \quad (30)$$

Note that the sign in (30) is proper: it takes 3 row exchanges to move the rows $k/3$, $2k/3$ and k to rows 1, 2, 3, and then $k/3-4$ exchanges to move the original row 1, which now is in row $k/3$, back to row 4, $2k/3-5$ exchanges to move the original row 2, which is now in row $2k/3$, back to row 5, and $k-6$ exchanges to move the original row 3, which is now in row k, back to row 6. The Vandermonde determinant in (30) is $= (-1)\beta_1^{k/3-1}\beta_2^{k/3-1}\beta_3^{k/3-1}(e_1 - e_2)(e_1-e_3)(e_2-e_3)$ which by (27) for $k-3$ and exponents $e_4, \ldots, e_k$

proves (27) for k. The leading monomial formulas (28, 29) are proven similarly. □

The remainder of Section 2, performing early termination (Item 1) or having a bound $B \geq t$ (Item 2), carries over. Theorem 4.1 generalizes to derivative of order ≥ 3 via its proof.

5 VARIANTS

The Prony algorithm for sparse polynomial interpolation has been used in many settings. One of the first is the multivariate version in [4]. In [24] the algorithm was deployed for Chebyshev and Pochhammer polynomial bases, which in [12] was generalized to Chebyshev bases of Second, Third and Fourth Kind. The numerical conditioning and stability has been studied extensively, as the algorithm builds exponential and harmonic sparse function approximations (see [6] and the references given). Digital error correction, that is, removing catastrophically false values, which is different from denoising the data, was introduced for the Prony algorithm in [5, 17], and for orthogonal bases in [2]. The state-of-the art for error correction in sparse univariate interpolation today is [20]. All those adaptations can be transferred to sparse Hermite interpolation.

5.1 Multivariate Sparse Hermite Interpolation

The Prony algorithm is transferred to multivariate polynomials in [4]. We explain the transfer to sparse interpolation with values of partial derivatives, that on the case of bivariate polynomials:

$$f(x_1,x_2) = \sum_{j=1}^{t} c_j x_1^{e_{j,1}} x_2^{e_{j,2}} \in \mathsf{K}[x_1,x_2], \qquad \forall j\colon c_j \neq 0, (e_{j,1},e_{j,2}) \in \mathbb{Z}^2.$$

The corresponding values are

$$a_i = f(\omega_1^{i_0+i}, \omega_2^{i_0+i}) \text{ and } \bar{a}_i = \omega_1^{i_0+i}\Big(\frac{\partial f(x_1,x_2)}{\partial x_1}\Big|_{x_1=\omega_1^{i_0+i}, x_2=\omega_2^{i_0+i}}\Big).$$

Again, the sequence $(\left[\begin{smallmatrix} a_i \\ \bar{a}_i \end{smallmatrix}\right])_{i\geq 0}$ is linearly generated by the term locator polynomial $\Lambda(z) = \prod_{j=1}^{t}(z - \omega_1^{e_{j,1}}\omega_2^{e_{j,2}})$ and for randomly selected ω_1, ω_2 the matrix $H_{t,r}$ (3) is non-singular with high probability, as indicated in the proof of Theorem 3.1. The technique in Step 4 in Algorithm 2.1 can possibly determine the $e_{j,1}$'s without a logarithm, but a logarithm computation is required for the $e_{j,2}$'s like in Steps 4(b)i and 4(b)ii. For fields K of characteristic $p > 0$, one may need an additional algorithm for computing $e_{j,1}, e_{j,2}$ from $b_j = \omega_1^{e_{j,1}}\omega_2^{e_{j,2}}$: in [4] ω_1, ω_2 are chosen different prime numbers for $\mathsf{K} = \mathbb{C}$ so that the b_j's are distinct and the $e_{j,1}, e_{j,2}$'s are recoverable, but for fields K of characteristic 0 we can obtain $e_{j,1}$ directly from the derivative values and $e_{j,2}$ as a logarithm. In summary, the sparse multivariate Hermite interpolation algorithm recovers $f(x_1,x_2)$ from $3t$ values, which include values at a partial derivative in one of the variables, at $t + \lceil t/2 \rceil$ distinct arguments for f.

Last, we note that the values at both partial derivatives may decrease the required number of arguments for certain exponents $e_{j,1}, e_{j,2}$, but for the polynomial $f(x,y) = \sum_{j=1}^{t} c_j (x_1 x_2)^{e_j}$, additional values of $\partial f(x_1,x_2)/\partial x_2$ seem not to help (cf. (26)). However, the algorithm in [27] transfers and partial derivatives in each variable can reduce the number of arguments to f in that algorithm, at the cost of more values. In general, the algorithm in [27] uses more values than the multivariate Prony algorithm (see also [9]).

5.2 Chebyshev Polynomial Basis

We explain the sparse recovery on Chebyshev polynomials of the First Kind; Second to Fourth Kind follow from the substitutions in [12]: Let $f(x)$ be a t-sparse polynomial in Chebyshev basis,

$$f(x) = \sum_{j=1}^{t} c_j T_{e_j}(x) \in \mathsf{K}[x], \forall j\colon c_j \neq 0, e_j \in \mathbb{Z}, e_1 > \cdots > e_t \geq 0. \quad (31)$$

where $(T_m)_{m\geq 0}$ are the m-degree Chebyshev Polynomials of the First Kind (we write "Chebyshev-1 Polynomials"):

$$\begin{bmatrix} T_m(x) \\ T_{m+1}(x) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2x \end{bmatrix}^m \begin{bmatrix} 1 \\ x \end{bmatrix}$$
$$\forall m \in \mathbb{Z};\ T_m\Big(\frac{1}{2}\big(y + \frac{1}{y}\big)\Big) = \frac{1}{2}\big(y^m + \frac{1}{y^m}\big);$$
$$T_m(\cos(\theta)) = \cos(m\theta) \text{ with } y = \exp(\mathrm{i}\theta). \quad (32)$$

Note that for negative subscripts we have $T_{-m}(x) = T_m(x)$. The recurrence (32) is non-trivial if the characteristic of K is $\neq 2$.

We suppose that both $f(x)$ and $f'(x)$ can be evaluated as a black box. The degrees e_j and coefficients c_j are recovered by sparse polynomial Hermite interpolation of $g(y)$ via the substitution

$$g(y) \stackrel{\text{def}}{=} f\Big(\frac{1}{2}\big(y + \frac{1}{y}\big)\Big) = \sum_{j=1}^{t} \frac{c_j}{2}\big(y^{e_j} + \frac{1}{y^{e_j}}\big) \quad [2]. \quad (33)$$

The Laurent polynomial $g(y)$ in power basis is a $2t$-sparse or a $(2t-1)$-sparse polynomial, the latter if $e_t = 0$, and algorithms of Section 2 apply. By the chain rule for the derivative we obtain

$$g'(y) = \frac{\mathrm{d}g(y)}{\mathrm{d}y} = \big(f'(x)\big|_{x=\frac{1}{2}(y+\frac{1}{y})}\big) \times \frac{1}{2}\big(1 - \frac{1}{y^2}\big),$$

which is a polynomial identity over any field of characteristic $\neq 2$, and therefore we have $\bar{g}(y) \stackrel{\text{def}}{=} y\, g'(y)$ is equal

$$\frac{1}{2}\big(f'(x)\big|_{x=\frac{1}{2}(y+\frac{1}{y})}\big)\big(y - \frac{1}{y}\big) = \sum_{j=1}^{t} \frac{c_j}{2}\big(e_j y^{e_j} + (-e_j)\frac{1}{y^{e_j}}\big).$$

If, for example, the Laurent polynomial $g(y)$ is $2t$-sparse, the sparse polynomial Hermite interpolation algorithm requires $6t$ values at $3t$ arguments. But because $g(y) = g(1/y)$ and $\bar{g}(y) = -\bar{g}(1/y)$ one only computes $3t + O(1)$ values of f: for example, we can set $i_0 = -\lceil (3t-1)/2 \rceil$ in Algorithm 2.1 and use $g(\omega^{i_0+i})$ and $\bar{g}(\omega^{i_0+i})$ for $i = 0, 1, \ldots, 3t-1$. Because the values for exponents $i_0 + i \geq 1$ can be computed from those at exponents $-i_0 - i$, one requires $N = 2(\lceil (3t-1)/2 \rceil + 1) \leq 3t + 2$ values of f and f' at $n = \lceil (3t-1)/2 \rceil + 1 \leq t + \lfloor t/2 \rfloor + 1$ arguments. Note that the additional "$+O(1)$" evaluations may be reduced by the technique in [12, Section 2], which uses the fact that the term locator polynomial Λ here is a reciprocal polynomial with a symmetric coefficient vector.

5.3 Error Correction

The error correction variants remove erroneous $\hat{a}_\kappa \neq f(\xi_\kappa) = a_\kappa$ and erroneous $\hat{\bar{a}}_\kappa \neq \xi_\kappa f'(\xi_\kappa) = \bar{a}_\kappa$ from the list of input values $\hat{a}_i$ for $i = 0, \ldots, L-1$, $\hat{\bar{a}}_i$ for $i = 0, \ldots, \bar{L}-1$, where B is an upper bound for the sparsity: $t \leq B$. Note that by the hats "$\hat{}$" on the input values we indicate that some values can be erroneous. As in the multiplicity code one assumes that there are $k \leq E$ errors, where the bound E is input and can be determined from the error rate

in the values. Furthermore, the algorithms, which are restricted to polynomial-time running time, produce a list of valid sparse Hermite interpolants. The basic algorithm uses $E+1$ redundant blocks of $3B$ values: we list the true polynomial values without errors:

$$\left.\begin{array}{cccc} f(\omega_1^{i_0}), & \ldots, f(\omega_1^{i_0+B+\lceil B/2\rceil-1}), & f'(\omega_1^{i_0}), & \ldots, f'(\omega_1^{i_0+B+\lfloor B/2\rfloor-1}), \\ \vdots & \vdots & \vdots & \vdots \\ f(\omega_{E+1}^{i_E}), \ldots, & f(\omega_{E+1}^{i_E+B+\lceil B/2\rceil-1}), & f'(\omega_{E+1}^{i_E}), \ldots, & f'(\omega_{E+1}^{i_E+B+\lfloor B/2\rfloor-1}). \end{array}\right\} \quad (34)$$

We enforce in (34) that all arguments are distinct: $\omega_\nu^i \neq \omega_\mu^m$ for $(\nu,i) \neq (\mu,m)$. If in the $N = 3(E+1)B$ input values $\hat{a}_i, \hat{\hat{a}}_i$ there are $k \leq E$ errors, at least one block is error-free and Algorithm 2.1 recovers f in the list of Hermite interpolants with $\leq E$ errors; there may be others. Here the sparsity is computed from the largest non-singular $H_{t,\lceil t/2\rceil}$ (3) of that block, and all remaining values and all values in all other blocks are checked to verify that there are $\leq E$ errors in total. We pursue the goal in [17, 20] and reduce the number of values $N < 3(E+1)B$ while recovering f in a list of Hermite interpolants, all of which have sparsity $\leq B$ and $\leq E$ errors in the values. Note that there are $n = (E+1)(B+\lceil B/2\rceil)$ distinct arguments to f in (34), and our goal is to have error correction at fewer arguments than without derivatives [17, 20]. The techniques in [17] are different from those in [20], the latter producing a higher error tolerance. Both methods apply to the Hermite setting.

As in [17], we can sub-sample at all arithmetic sequences of exponents: $\hat{a}_{\chi+i\psi}$ for $i = 0, \ldots, B+\lceil B/2\rceil - 1$, $\hat{\hat{a}}_{\chi+i\psi}$ for $i = 0, \ldots, B+\lfloor B/2\rfloor - 1$. For a list of $N = 2n$ values $\hat{a}_{i_0+i}, \hat{\hat{a}}_{i_0+i}$, $i = 0, 1, \ldots, n-1$, we guarantee that there are no errors for one pair (χ, ψ). On that sub-sample the t-sparse polynomial $f_{\chi,\psi}(x) \stackrel{\text{def}}{=} \sum_{j=1}^t c_j \omega^{e_j\chi} x^{e_j\psi}$ is interpolated. We have $\bar{a}_{\chi+i\psi} = \sum_{j=1}^t c_j e_j \omega^{e_j(\chi+i\psi)}$,

$$\omega^i f'_{\chi,\psi}(\omega^i) = \omega^i \sum_{j=1}^t c_j \omega^{e_j\chi} e_j \psi \omega^{(e_j\psi-1)i} = \psi \bar{a}_{\chi+i\psi}. \quad (35)$$

Therefore, Algorithm 2.1 recovers $f_{\chi,\psi}(x)$ from the good values $a_{\chi+i\psi}$ and $\psi\bar{a}_{\chi+i\psi}$. From (χ,ψ) and $f_{\chi,\psi}(x)$ we compute f. Finally, we determine the minimum $n_{B+\lceil B/2\rceil,E}$ such that in the sequence of 2-dimensional vectors $\begin{bmatrix}\hat{a}_i \\ \hat{\hat{a}}_i\end{bmatrix}$ for $i = 0, \ldots, n_{B+\lceil B/2\rceil,E} - 1$ one guarantees a sub-sample $\begin{bmatrix}\hat{a}_{\chi+\psi i} \\ \hat{\hat{a}}_{\chi+\psi i}\end{bmatrix}$ for $i = 0, \ldots, B+\lceil B/2\rceil - 1$ without any of the E errors. For example, we obtain from [17, Table 1], that when removing any arbitrary 11 elements from $\{1, \ldots, 124\}$ one always has an arithmetic sub-sequence of length 12 in the remaining elements. Therefore, for $B = 8, E = 11$ and $n = n_{12,11} = 124$ we can always find a sub-sample of $B + B/2 = 12$ clean value vectors from which one can interpolate the 8-sparse polynomial f. Overall, we need $N = 2n = 2 \cdot 124 = 248$ values to correct $\leq E = 11$ errors; note that $3(E+1)B = 3 \cdot 12 \cdot 8 = 288$, which is the count (34) for redundant blocking. Without derivatives, one needs $N = n = n_{16,11}$ values. The exact entry is not computed in [17, Table 1], but we have $n_{16,11} > n_{13,11} = 156$, which is > 124 arguments.

In [20] we raise the decodable error rate by a different redundant blocking technique. We demonstrate the technique on the example $B = 3$ (cf. [20, Example 1]). We first interpolate $\hat{a}_i$ for $i = 0, \ldots, 6 = 2B$ and $\hat{\hat{a}}_i$ for $i = 0, \ldots, 4 = 2B-2$, a total of $4B = 12$ values, and correct $\leq E = 1$ error. The augmented coefficient matrix that corresponds to the linear system (3) is

$$\begin{array}{ll} \text{Row 1} & \begin{bmatrix} \hat{a}_0 & \hat{a}_1 & \hat{a}_2 & \hat{a}_3 \\ \hat{a}_1 & \hat{a}_2 & \hat{a}_3 & \hat{a}_4 \\ \hat{a}_2 & \hat{a}_3 & \hat{a}_4 & \hat{a}_5 \\ \hat{a}_3 & \hat{a}_4 & \hat{a}_5 & \hat{a}_6 \\ \hat{\hat{a}}_0 & \hat{\hat{a}}_1 & \hat{\hat{a}}_2 & \hat{\hat{a}}_3 \\ \hat{\hat{a}}_1 & \hat{\hat{a}}_2 & \hat{\hat{a}}_3 & \hat{\hat{a}}_4 \end{bmatrix} \end{array} \begin{array}{l} \text{Row 1} \\ \text{Row 2} \\ \text{Row 3} \\ \text{Row 4} \\ \text{Row 5} \\ \text{Row 6} \end{array}. \quad (36)$$

Again, with ≤ 1 erroneous entry, we recover f and possibly other interpolants that cause ≤ 1 error. If the error is in the $\hat{\hat{a}}_i$, we compute the candidate interpolant from the first 4 rows by the Prony algorithm. If the error is in $\hat{a}_2 \neq f(\omega^{i_0+2})$, say, we compute candidate interpolants from Rows 4–6. We have for $b_1 = \omega^{e_1}$, $b_2 = \omega^{e_2}$, $b_2 = \omega^{e_2}$:

$$\begin{bmatrix} a_3 & a_4 & a_5 \\ \bar{a}_0 & \bar{a}_1 & \bar{a}_2 \\ \bar{a}_1 & \bar{a}_2 & \bar{a}_3 \end{bmatrix} = \begin{bmatrix} b_1^3 & b_2^4 & b_3^5 \\ e_1 & e_2 & e_3 \\ e_1 b_1 & e_2 b_2 & e_3 b_3 \end{bmatrix} \begin{bmatrix} c_1 b_1^{i_0} & 0 & 0 \\ 0 & c_2 b_2^{i_0} & 0 \\ 0 & 0 & c_3 b_3^{i_0} \end{bmatrix} \begin{bmatrix} 1 & b_1 & b_1^2 \\ 1 & b_2 & b_2^2 \\ 1 & b_3 & b_3^2 \end{bmatrix} \quad (37)$$

(cf. (10)). As in Theorem 3.1 and Corollary 3.1, for randomly selected ω the 3×3 matrix in (37) is non-singular with high probability and f is computed as one among the valid Hermite interpolants. Similarly, we handle erroneous $\hat{a}_0, \hat{a}_1, \hat{a}_4, \hat{a}_5$ and $\hat{a}_6$. If $\hat{a}_3$ is erroneous, we proceed as in [20]: we replace $\hat{a}_3$ by a variable α and compute the determinant of the 4×4 submatrix of the first 4 rows. We obtain a degree 4 polynomial in α one of whose roots is the value a_3. Therefore, we can try all 4 roots in computing the candidate interpolants.

Now suppose that we wish to correct E errors. The count of values is

$$N = \left(E + 2\left\lceil \frac{E-1}{2} \right\rceil + 3\right) B, \; n = \begin{cases} \frac{E}{2}(2B+1) + B + \lceil \frac{B}{2} \rceil & \text{if } E \text{ even}, \\ \frac{E+1}{2}(2B+1) & \text{if } E \text{ odd}. \end{cases} \quad (38)$$

First, we assume that $E \equiv 1 \pmod 2$. We interpolate values in $(E+1)/2$ blocks of $4B = 12$ values. There cannot be ≥ 2 errors in each block, because then there would be $E+1$ errors. Therefore, one block has ≤ 1 error, and the algorithm computes f among the candidate interpolants. In summary, from a total of $N = (2E+2)B = (E+1)/2\ (4B)$ values we can list-Hermite-interpolate a polynomial of sparsity $\leq B$ in the presence of $\leq E$ errors. Second, if $E \equiv 0 \pmod 2$, the total count is $N = E/2\ (4B) + 3B$, which are $E/2$ blocks with $4B$ values and one block with $3B$ values. One can decode from the error-free last block. If there are ≥ 1 errors in the last block, there must be one block of $4B = 12$ values with ≤ 1 error, which interpolates f. For $B = 8$ and $E = 11$ we have $N = 200$ values at $n = 102$ arguments, both of which are fewer than the counts $N = 248$ and $n = 124$ above by sub-sampling at arithmetic sub-sequences of indices. The comparable count in [20] without derivatives is $N = n = \lfloor 4E/3 + 2\rfloor B = 128$, which is fewer values at > 102 arguments.

6 CONCLUSION

In the Prony approach to sparse model construction, the gradient (rate of change) of the data at a measurement point can be used to reduce the number of measurement points, but that at the cost of more overall data. A main difference is that the minimum number of measurement points is achieved only if the points incorporate some randomness. The precise mathematical nature of those non-random points for which over-sampling becomes necessary and how much oversampling suffices remains to be understood.

ACKNOWLEDGMENTS

I thank the reviewers for their comments.

This research was supported by the National Science Foundation under Grant CCF-1717100.

Note added on August 29, 2022: Added reference [25] at end of Section 1.1.

Note added on February 4, 2024: Correct (37).

REFERENCES

[1] Ben Adcock and Yi Sui. 2019. Compressive Hermite Interpolation: Sparse, High-Dimensional Approximation from Gradient-Augmented Measurements. *Constructive Approximation* 50 (2019), 167–207. Issue 1. URL: https://doi.org/10.1007/s00365-019-09467-0, https://arxiv.org/abs/1712.06645.

[2] Andrew Arnold and Erich L. Kaltofen. 2015. Error-Correcting Sparse Interpolation in the Chebyshev Basis, See [13], 21–28. URL: http://users.cs.duke.edu/~elk27/bibliography/15/ArKa15.pdf.

[3] Andrew Arnold and Daniel S. Roche. 2015. Output-sensitive algorithms for sumset and sparse polynomial multiplication, See [13], 29–36. URL: https://arxiv.org/abs/1501.05296.

[4] M. Ben-Or and P. Tiwari. 1988. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proc. Twentieth Annual ACM Symp. Theory Comput.* ACM Press, New York, N.Y., 301–309.

[5] Matthew T. Comer, Erich L. Kaltofen, and Clément Pernet. 2012. Sparse Polynomial Interpolation and Berlekamp/Massey Algorithms That Correct Outlier Errors in Input Values. In *ISSAC 2012 Proc. 37th Internat. Symp. Symbolic Algebraic Comput.*, Joris van der Hoeven and Mark van Hoeij (Eds.). Association for Computing Machinery, New York, N. Y., 138–145. URL: http://users.cs.duke.edu/~elk27/bibliography/12/CKP12.pdf.

[6] Annie Cuyt and Wen-shin Lee. 2020. How to get high resolution from sparse and coarsely sampled data. *Applied Comput. Harmonic Analysis* 48 (May 2020), 1066–1087. Issue 3. URL: https://doi.org/10.1016/j.acha.2018.10.001.

[7] Mark Giesbrecht, George Labahn, and Wen-shin Lee. 2009. Symbolic-numeric sparse interpolation of multivariate polynomials. *J. Symbolic Comput.* 44 (2009), 943–959.

[8] Zhiwei Hao, Erich L. Kaltofen, and Lihong Zhi. 2016. Numerical Sparsity Determination and Early Termination. In *ISSAC'16 Proc. 2016 ACM Internat. Symp. Symbolic Algebraic Comput.*, Markus Rosenkranz (Ed.). Association for Computing Machinery, New York, N. Y., 247–254. URL: http://users.cs.duke.edu/~elk27/bibliography/16/HKZ16.pdf.

[9] J. van der Hoeven and G. Lecerf. 2015. Sparse polynomial interpolation in practice. *ACM Commun. Comput. Algebra* 48, 3/4 (2015), 187–191. URL: http://hal.archives-ouvertes.fr/hal-00980366.

[10] Qiao-Long Huang. 2019. Sparse Polynomial Interpolation over Fields with Large or Zero Characteristic. In *Proceedings of the 2019 on International Symposium on Symbolic and Algebraic Computation, ISSAC 2019, Beijing, China, July 15-18, 2019*, James H. Davenport, Dongming Wang, Manuel Kauers, and Russell J. Bradford (Eds.). ACM, 219–226. URL: https://doi.org/10.1145/3326229.3326250.

[11] Qiao-Long Huang. 2020. Sparse Polynomial Interpolation Based on Derivative. *CoRR* abs/2002.03708 (2020). arXiv:2002.03708 URL https://arxiv.org/abs/2002.03708.

[12] Erdal Imamoglu, Erich L. Kaltofen, and Zhengfeng Yang. 2018. Sparse Polynomial Interpolation With Arbitrary Orthogonal Polynomial Bases. In *ISSAC '18 Proc. 2018 ACM Internat. Symp. Symbolic Algebraic Comput.*, Carlos Arreche (Ed.). Association for Computing Machinery, New York, N. Y., 223–230. In memory of Bobby F. Caviness (3/24/1940–1/11/2018). URL: http://users.cs.duke.edu/~elk27/bibliography/18/IKY18.pdf.

[13] ISSAC 2015 2015. *ISSAC'15 Proc. 2015 ACM Internat. Symp. Symbolic Algebraic Comput.* Association for Computing Machinery, New York, N. Y.

[14] Erich Kaltofen. 2021. Hermite Interpolation With Error Correction. Invited lecture at ACA 2021, Session on Hybrid Symbolic-Numeric Computation. URL: https://users.cs.duke.edu/~elk27/bibliography/21/ACA2021-Abstract-Kaltofen.pdf.

[15] E. Kaltofen and Lakshman Yagati. 1988. Improved sparse multivariate polynomial interpolation algorithms. In *Symbolic Algebraic Comput. Internat. Symp. ISSAC '88 Proc. (Lect. Notes Comput. Sci., Vol. 358)*, P. Gianni (Ed.). Springer Verlag, Heidelberg, Germany, 467–474. URL: http://users.cs.duke.edu/~elk27/bibliography/88/KaLa88.pdf.

[16] Erich Kaltofen and Wen-shin Lee. 2003. Early Termination in Sparse Interpolation Algorithms. *J. Symbolic Comput.* 36, 3–4 (2003), 365–400. Special issue Internat. Symp. Symbolic Algebraic Comput. (ISSAC 2002). Guest editors: M. Giusti & L. M. Pardo. URL: http://users.cs.duke.edu/~elk27/bibliography/03/KL03.pdf.

[17] Erich L. Kaltofen and Clément Pernet. 2014. Sparse Polynomial Interpolation Codes and Their Decoding Beyond Half the Minimal Distance. In *ISSAC 2014 Proc. 39th Internat. Symp. Symbolic Algebraic Comput.*, Katsusuke Nabeshima (Ed.). Association for Computing Machinery, New York, N. Y., 272–279. URL: http://users.cs.duke.edu/~elk27/bibliography/14/KaPe14.pdf.

[18] Erich L. Kaltofen, Clément Pernet, and Zhi-Hong Yang. 2020. Hermite Rational Function Interpolation with Error Correction. In *Computer Algebra in Scientific Computing, CASC 2020 (Lect. Notes Comput. Sci., Vol. 12291)*, F. Boulier, M. England, T. Sadykov, and E. Vorozhtsov (Eds.). Springer, 335–357. URL: http://users.cs.duke.edu/~elk27/bibliography/20/KPY20.pdf, https://doi.org/10.1007/978-3-030-60026-6_19.

[19] Erich L. Kaltofen, Clément Pernet, and Zhi-Hong Yang. 2021. Hermite Interpolation With Error Correction: Fields of Zero or Large Characteristic and Large Error Rate. In *ISSAC '21 Proc. 2021 ACM Internat. Symp. Symbolic Algebraic Comput.*, Marc Mezzarobba (Ed.). Association for Computing Machinery, New York, N. Y., 241–247. URL: http://users.cs.duke.edu/~elk27/bibliography/21/KPY21.pdf, https://doi.org/10.1145/3452143.3465525.

[20] Erich L. Kaltofen and Zhi-Hong Yang. 2021. Sparse Interpolation With Errors in Chebyshev Basis Beyond Redundant-Block Decoding. *IEEE Trans. Information Theory* 67, 1 (Jan. 2021), 232–243. URL: http://users.cs.duke.edu/~elk27/bibliography/19/KaYa19.pdf, https://ieeexplore.ieee.org/document/9207761, https://arxiv.org/abs/1912.05719.

[21] Erich Kaltofen and George Yuhasz. 2013. A Fraction Free Matrix Berlekamp/Massey Algorithm. *Linear Algebra and Applications* 439, 9 (Nov. 2013), 2515–2526. URL: http://users.cs.duke.edu/~elk27/bibliography/08/KaYu08.pdf.

[22] Erich Kaltofen and George Yuhasz. 2013. On The Matrix Berlekamp-Massey Algorithm. *ACM Trans. Algorithms* 9, 4 (Sept. 2013). URL: http://users.cs.duke.edu/~elk27/bibliography/06/KaYu06.pdf.

[23] Swastik Kopparty. 2015. List-decoding multiplicity codes. *Theory of Computing* 11, 1 (2015), 149–182. URL: https://sites.math.rutgers.edu/~sk1233/part2.pdf.

[24] Lakshman Y. N. and B. D. Saunders. 1995. Sparse polynomial interpolation in non-standard bases. *SIAM J. Comput.* 24, 2 (1995), 387–397.

[25] Thomas Peter, Gerlind Plonka, and Daniela Roşca. 2013. Representation of sparse Legendre expansions. *J. Symbolic Comput.* 50 (March 2013), 159–169. URL: https://www.sciencedirect.com/science/article/pii/S0747717112001101.

[26] R. Prony. III (1795). Essai expérimental et analytique sur les lois de la Dilatabilité de fluides élastiques et sur celles de la Force expansive de la vapeur de l'eau et de la vapeur de l'alkool, à différentes températures. *J. de l'École Polytechnique* 1 (Floréal et Prairial III (1795)), 24–76. R. Prony is Gaspard(-Clair-François-Marie) Riche, baron de Prony.

[27] R. Zippel. 1990. Interpolating polynomials from their values. *J. Symbolic Comput.* 9, 3 (1990), 375–403.

7 APPENDIX: PROOF OF THEOREM 3.1 DETAILS

As stated, the formulas are proven by induction with minor expansion along the first column of $W_{k,r}$ in (11). The signs σ and exponents η satisfy the following recurrences:

$$\sigma(k,r) = \begin{cases} 1 & \text{for } k{=}1 \text{ and } (r{=}0 \text{ or } r{=}1), \\ -1 & \text{for } (k,r) = (2,1), \\ (-1)^{r+1}\sigma(k{-}1,r{-}1) & \text{for } \frac{k}{2} < r \le k, \\ (-1)^{k+1}\sigma(k{-}1,r) & \text{for } 0 \le r < \frac{k}{2}, \\ (-1)^{k/2}\sigma(k{-}2,\frac{k}{2}{-}1) & \text{for } k{\ge}4 \text{ and } k{\equiv}0 \pmod 2 \\ & \qquad \text{and } r{=}k/2; \end{cases} \quad (39)$$

$$\eta(k,r,j) = \left\{ \begin{array}{ll} k{-}j & \text{for } r{=}0 \text{ or } r{=}k, \\ r{-}1 & \text{for } 2r{>}k \text{ and } j{=}1, \\ \eta(k{-}1,r{-}1,j{-}1) & \text{for } 2r{>}k \text{ and } j{>}1, \\ k{-}r{-}1 & \text{for } 2r{<}k \text{ and } j{=}1, \\ \eta(k{-}1,r,j{-}1) & \text{for } 2r{<}k \text{ and } j{>}1, \\ \frac{k}{2}{-}\lceil\frac{j}{2}\rceil & \text{for } 2r{=}k. \end{array} \right\} \quad (40)$$

For $k = 1$ one has $\det(W_{1,1}) = \det(\begin{bmatrix}1\end{bmatrix}) = 1$ and $\det(W_{1,0}) = \det(\begin{bmatrix}e_1\end{bmatrix}) = e_1$. Now let $k \ge 2$. We use the exponents $e_1, \ldots, e_k$ as arguments to $W_{k,r}(e_1, \ldots, e_k)$ in (11). Note that here $\beta_j = x^{e_j}$; one could also use a multivariate term order with the variable order $\beta_1 > \cdots > \beta_k$, as we need in Section 5.1. If $r - 1 > k - r - 1$, then β_1^{r-1} is the highest degree term in the first column, and the

leading monomial of $\det(W_{k,r}(e_1,\ldots,e_k))$ is $(-1)^r\beta_1^{r-1}\times$ the leading monomial of $\det(W_{k-1,r-1}(e_2,\ldots,e_k))$. If $r-1 < k-r-1$, then β_1^{k-r-1} is the highest degree term in the first column, and the leading monomial of $\det(W_{k,r}(e_1,\ldots,e_k))$ is $(-1)^{k-r}e_1\beta_1^{k-r-1}$ $\times$ the leading monomial of $\det(W_{k-1,r}(e_2,\ldots,e_k))$. We obtain (12, 39, 40) for all $k \neq 2r$.

Finally, we establish the formulas for $k = 2r$. The leading monomial, denoted by LM, of $\det(W_{k,k/2}(e_1,\ldots,e_k))$ is

$$= (-1)^{k-1}e_1\beta_1^{k/2-1}\times \mathrm{LM}\big(\det(W_{k-1,k/2}(e_2,\ldots,e_k))\big)+ \quad (41)$$

$$(-1)^{k/2-1}\beta_1^{k/2-1}\times \mathrm{LM}\big(\det(W_{k-1,k/2-1}(e_2,\ldots,e_k))\big). \quad (42)$$

The leading monomial in (41) is given by (16) and the leading monomial in (42) is given by (17), both for $k-1$ replacing k. Applying the induction hypothesis for $k-1$, one obtains as the leading monomial of $W_k(e_1,\ldots,e_k)$, using (39):

$$\Big((-1)\times\underbrace{\sigma(k-1,\frac{k}{2})}_{(-1)^{k/2+1}\sigma(k-2,\frac{k}{2}-1)}e_1+(-1)^{k/2-1}\underbrace{\sigma(k-1,\frac{k}{2}-1)}_{(-1)^k\sigma(k-2,\frac{k}{2}-1)}e_2\Big)\,\beta_1^{k/2-1}\times$$

$$(e_3-e_4)\cdots(e_{k-1}-e_k)\beta_2^{k/2-1}(\beta_3\beta_4)^{k/2-2}\cdots\beta_{k-3}\beta_{k-2}. \quad (43)$$

Because k is even, the first factor in (43) is $=\sigma(k,k/2)(e_1-e_2)$ by (39), which concludes the inductive proof.

8 APPENDIX

This appendix is not included in the ISSAC Proceedings.

Notation (in alphabetic order):	
A_i	$A_i = \left[\begin{smallmatrix} f(\omega^{i_0+i}) \\ \omega^{i_0+i}f'(\omega^{i_0+i}) \end{smallmatrix}\right] = \left[\begin{smallmatrix} a_i \\ \bar{a}_i \end{smallmatrix}\right] \in \mathsf{K}^2$ for $i = 0, 1, \ldots$ (5)
$a_i, \bar{a}_i, \bar{\bar{a}}_i$	$a_i = f(\omega^{i_0+i})$, $\bar{a}_i = \omega^{i_0+i}f'(\omega^{i_0+i})$ (Section 1 and Input to Algorithm 2.1), $\bar{\bar{a}}_i = \omega^{2(i_0+i)}f''(\omega^{i_0+i})$ (Section 4)
$\hat{a}_i, \hat{\bar{a}}_i$	either $a_i, \bar{a}_i$ or an erroneous value (Section 5.3)
B	a term bound $B \geq t$ (Item 2)
b_j	the term values $b_j = \omega^{e_j}$
β_j	the terms $\beta_j = x^{e_j}$ or $\beta_j = x_1^{e_{j,1}}x_2^{e_{j,2}}$
c_j	the coefficients of the terms in f
D	an upper bound $D \geq \deg(f)$ (Item 1)
δ	the discrepancy (8)
E	an upper bound for the number of errors in $\hat{a}_i, \hat{\bar{a}}_i$ (Section 5.3)
e_j	the term exponents in f
$f(x)$	the sparse interpolant (1)
$g(y)$	the transformed $f((y+1/y)/2)$ (33)

Notation continued (in alphabetic order):	
θ	the order of ω: $\omega^\theta = 1$ (Case 2 in the Output specifications of Algorithm 2.1)
$H_{t,r}, H_{t,r_0,r_1}$	the coefficient matrices (3) and (25), resp.
$\mathcal{H}_{t,r}$	the symbolic coefficient matrix (10)
$h(x)$	an alternate sparse interpolant if ω is unlucky; see last paragraph of Section 2
$\eta(k,r,j)$	the exponents of the leading term of $\det(W_{t,r})$ (14,40)
i_0, i	indices of the arguments ξ_i, ω^{i_0+i} of f, f'
j	indices of the terms in f
K	the field of scalars
k	the intermediate dimensions of $W_{k,r}$ (11), $W_{k,r}^{[j^*]}$ (18), W_{t,r_0,r_1} (26); the actual number of errors $k \leq E$ (Section 5.3)
ℓ	the order of derivatives $f^{(\ell)}$ for which there are values
$\lambda_j, \Lambda(z)$	the term locator polynomial $\Lambda(z) = z^t+\lambda_{t-1}z^{t-1}+\cdots+\lambda_0 = \prod_{j=1}^t(z-\omega^{e_j})$ (2).
N	the total number of values, including values at derivatives
n	the number of arguments ξ_i, ω^{i_0+i}
ξ_i	arguments to f, f'
p	the characteristic of K, if positive
r, r_0, r_1	the number of distinct arguments to f is $t+r$ (3) or $t+r_0$ (25), to f' it is $2t-r$ (3) or $t+r_1$ (25)
S	the finite subset $S \subseteq \mathsf{K}$ from which ω is sampled
$\sigma(k,r)$	the sign of the leading term of $\det(W_{t,r})$ (13,39)
$T_m(x)$	the m-degree Chebyshev polynomial of the First Kind (Section 5.2)
t	the number of terms in f
$\tau(k,r,j^*,j)$	row permutation on $W_{t,r}$ to move rows with $e_j = 0$ back (19)
$W_{t,r}, W_{t,r}^{[j^*]}$	the left matrix factor of $\mathcal{H}_{t,r}$ (10); the right factor is a transposed "V"andermonde matrix, hence the "W"
W_{t,r_0,r_1}	the symbolic left matrix factor for 2 derivatives (26)
x, x_1, x_2	the variables in f
χ, ψ	the linear arithmetic index sub-sequence $\chi+i\psi$ for sub-sampling (Section 5.3)
z	the variable in the term locator polynomial
ω	the randomly selected base for the arguments $\xi_i = \omega^{i_0+i}$