Sparse Polynomial Interpolation With Arbitrary Orthogonal Polynomial Bases

In memory of Bobby F. Caviness (3/24/1940–1/11/2018)

Erdal Imamoglu
Dept. of Math., NCSU
Raleigh, NC, USA

Erich L. Kaltofen
Dept. of Math., NCSU
Raleigh, NC, USA

Zhengfeng Yang
Key Lab Trustworthy Comput.
ECNU, Shanghai, China

ABSTRACT

An algorithm for interpolating a polynomial f from evaluation points whose running time depends on the sparsity t of the polynomial when it is represented as a sum of t Chebyshev Polynomials of the First Kind with non-zero scalar coefficients is given by Lakshman Y. N. and Saunders [SIAM J. Comput., vol. 24, nr. 2 (1995)]; Kaltofen and Lee [JSC, vol. 36, nr. 3–4 (2003)] analyze a randomized early termination version which computes the sparsity t. Those algorithms mirror Prony's algorithm for the standard power basis to the Chebyshev Basis of the First Kind. An alternate algorithm by Arnold's and Kaltofen's [Proc. ISSAC 2015, Sec. 4] uses Prony's original algorithm for standard power terms.

Here we give sparse interpolation algorithms for generalized Chebyshev polynomials, which include the Chebyshev Bases of the Second, Third and Fourth Kind. Our algorithms also reduce to Prony's algorithm. If given on input a bound $B \geq t$ for the sparsity, our new algorithms deterministically recover the sparse representation in the First, Second, Third and Fourth Kind Chebyshev representation from exactly $t + B$ evaluations.

Finally, we generalize our algorithms to bases whose Chebyshev recurrences have parametric scalars. We also show how to compute those parameter values which optimize the sparsity of the representation in the corresponding basis, similar to computing a sparsest shift.

ACM Reference Format:
Erdal Imamoglu, Erich L. Kaltofen, and Zhengfeng Yang. 2018. Sparse Polynomial Interpolation With Arbitrary Orthogonal Polynomial Bases: In memory of Bobby F. Caviness (3/24/1940–1/11/2018). In *ISSAC'18: 2018 ACM Int'l Symposium on Symbolic & Algebraic Computation, July 16–19, 2018, NY, NY, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3208976.3208999

1. INTRODUCTION

We consider the problem of reconstructing the term degrees and non-zero coefficients of a univariate polynomial f whose evaluation we can obtain at arbitrary values for the variable for a black box for the polynomial. Here f is represented in an orthogonal term basis $P_0(x), P_1(x), P_2(x), \ldots$

$$f(x) = \sum_{j=1}^{t} c_j P_{\delta_j}(x), c_j \in \mathsf{K}, c_j \neq 0, 0 \leq \delta_1 < \delta_2 < \cdots < \delta_t \quad (1)$$

where P_δ are Chebyshev Polynomials of the first, second, or third kind and where K is an arbitrary field of characteristic $\neq 2$. Our algorithms compute the term degrees δ_j and term coefficients c_j, hence perform a sparse polynomial interpolation with one of the Chebyshev bases. The main idea is to reduce the sparse interpolation problem in Chebyshev basis to a sparse interpolation problem in the power basis and apply Prony's algorithm [5, 23] (the 1959 Bose-Chaudhuri-Hocquenghem error correction decoding algorithm) to the latter problem.

As with Prony's algorithm, the sparsity of t need not be given on input. We consider two early termination strategies that determine t: if a bound $B \geq t$ is given on input, we compute t and f deterministically from $t + B$ evaluations. A difficulty is that the constructed Prony problem has sparsity $2t$ and we have to exploit its special structure to reduce the number of evaluations. Our deterministic algorithm can be implemented in $(t + B)^{2+o(1)}$ field operations, degree-t polynomial root finding, and computing t integer logarithms in K. The quadratic exponent is a consequence of the lack of fast main-diagonal Toeplitz solvers with arbitrary look-ahead (cf. [4, 6, 24]). For finite coefficient fields K with a fast discrete logarithm algorithm [21] our algorithm is of bit complexity $((t+B)\log(\deg f))^{O(1)}$. We can also compute t and f by Kaltofen's and Lee's randomized early termination strategy from $2t + 2$ evaluations (see [1, Sec. 4.2]). In order to use soft-linear randomized Toeplitz/Hankel solvers with $t^{1+o(1)}$ arithmetic operations [15] one needs to oversample to $2B$ or $2^{\lceil \log_2(2t+2) \rceil}$ evaluations, respectively.

We now recall the properties of the Chebyshev Polynomials of the First, Second, Third and Fourth Kind. Traditionally, those are n-degree polynomials in x over the field of real numbers denoted by $T_n(x)$ (First Kind), $U_n(x)$ (Second Kind), $V_n(x)$ (Third Kind) and $W_n(x)$ (Fourth Kind). If $P_n(x)$ denotes any of those four polynomials, we have

$$P_0(x) = 1, \quad P_n(x) = 2x\,P_{n-1}(x) - P_{n-2}(x) \text{ for } n \geq 2, \quad (2)$$

$P_n = T_n, U_n, V_n, W_n$, and the distinct initializations at $n = 1$,

$$T_1(x) = x, \quad U_1(x) = 2x, \quad V_1(x) = 2x - 1, \quad W_1(x) = 2x + 1. \quad (3)$$

An alternative to (2) is

$$\begin{bmatrix} P_n(x) \\ P_{n+1}(x) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2x \end{bmatrix}^n \begin{bmatrix} 1 \\ P_1(x) \end{bmatrix} \quad \text{for } n \in \mathbb{Z}. \quad (4)$$

Note that (4) extends the subscript range n to negative integers and by computing the power of 2×2 coefficient matrix gives an algorithm for evaluating all P_n in $O(\log(n))$ scalar operations. All four kinds yield a vector space basis for the ring of polynomials over any field K of characteristic $\neq 2$. From now on, we shall speak of Chebyshev-1, Chebyshev-2, Chebyshev-3 and Chebyshev-4 polynomials and bases in reference to first, second, third and fourth kind.

The third and fourth kind polynomials are not as common, because we have $W_n(x) = (-1)^n V_n(-x)$ and $V_n(x) = T_{2n+1}(z)/z$ and $W_n(x) = U_{2n}(z)$ for $z = \sqrt{(x+1)/2}$, that is, the two identities are stated in the algebraic function field $\mathsf{K}(x)[z]/(z^2 - (x+1)/2)$.

ISSAC'18, July 16–19, 2018, New York, NY, USA

ACM ISBN 978-1-4503-5550-6/18/07…$15.00
https://doi.org/10.1145/3208976.3208999

There are some well known properties that are the basis of sparse interpolation in Chebyshev-1 Basis.

FACT 1.1. *Let $m, n \in \mathbb{Z}_{\geq 0}$. Then the following hold:*

i. $T_n(T_m(x)) = T_{mn}(x) = T_m(T_n(x))$.

ii. $T_n(\frac{x+\frac{1}{x}}{2}) = \frac{x^n+\frac{1}{x^n}}{2}$ *for all* $n \geq 0$.

Based on Fact 1.1.i, which is that Chebyshev-1 Polynomials commute with respect to composition, Lakshman and Saunders [20] have mirrored Prony's algorithm in order to reconstruct the list of non-zero coefficients c_j and the list of corresponding degrees δ_j from evaluations of

$$f(x)=c_1T_{\delta_1}(x)+\cdots+c_tT_{\delta_t}(x), c_j\neq 0, 0\leq\delta_1<\delta_2<\cdots<\delta_t \quad (5)$$

at $x = T_0(\beta), T_1(\beta), \ldots$ for a scalar β (see also [1, 9, 19, Sec. 3]). Their reconstruction algorithm is thus a sparse interpolation algorithm in Chebyshev-1 Basis. For sparsity in Chebyshev-2 Polynomials U_n one obstruction is the lack of the commuting property of term substitution. However, performing the substitution given in Fact 1.1.ii, Arnold and Kaltofen [1, Sec. 4] directly reduced the sparse polynomial (5) to a sparse Laurent polynomial in power (standard) basis. More precisely, for f in (5) we have

$$g(y) \stackrel{\text{def}}{=} f((y+\tfrac{1}{y})/2) = \textstyle\sum_{j=1}^{t} \frac{c_j}{2}\,(y^{\delta_j}+y^{-\delta_j}) \quad (6)$$

and Prony's algorithm can reconstruct the sparse Laurent polynomial g. Here we use the corresponding properties to Fact 1.1.ii for U_n, V_n, W_n, namely,

$$(y-\tfrac{1}{y})\ U_n\Big((y+\tfrac{1}{y})/2\Big) = y^{n+1} - \tfrac{1}{y^{n+1}}, \quad (7)$$

$$(y+\tfrac{1}{y})\ V_n\Big((y^2+\tfrac{1}{y^2})/2\Big) = y^{2n+1} + \tfrac{1}{y^{2n+1}}, \quad (8)$$

$$(y-\tfrac{1}{y})\ W_n\Big((y^2+\tfrac{1}{y^2})/2\Big) = y^{2n+1} - \tfrac{1}{y^{2n+1}}. \quad (9)$$

Note that the multiplicative preconditioner $y \pm 1/y$ is introduced before interpolating the substituted $f((y+\frac{1}{y})/2)$ or $f((y^2+\frac{1}{y^2})/2)$, thus overcoming the long-known obstruction for sparse interpolation with a Chebyshev-2 Basis. Potts and Tasche [22, Equation 4.2] have introduced a corresponding trigonometric multiplier: $\sin(\alpha)\times U_n(\cos(\alpha)) = \sin((n+1)\alpha)$, which with $y = e^{\boldsymbol{i}\alpha}$ is (7). Our substitution does not require the evaluation of a transcendental function and can be realized as an exact algorithm even over a finite field, while the algorithm in [22] uses floating point arithmetic. We think of the polynomial $f(x)$ as a black box polynomial that can be arbitrarily probed. For Kaltofen and Lee [19] randomized sparse interpolation from $2t+2$ values with early termination, an upper bound of $\deg(f)$ is required on input for achieving success probability $\geq 1/2$, for otherwise the polynomials $\prod_j(x-\beta_j)$ and 0 are indistinguishable, where β_j ranges over all possible random choices of evaluation points. For our bases, see Theorem 5.2.

The Lakshman-Saunders [20] method and the Arnold-Kaltofen [1, Sec. 4] substitution (6), which is the approach also here, are related by the substitution $\beta = (\omega + 1/\omega)/2$ for the base points β and ω of the evaluations. That substitution has 2 effects: 1. the arising Toeplitz-plus-Hankel system in Lakshman-Saunders becomes a Toeplitz system; 2. the degrees of the terms are computed as logarithms with integral output values. The Toeplitz matrix allows for the use of the Berlekamp-Massey algorithm. The substitution (6) and (7–9) double the sparsity in the arising Laurent polynomial (an exception is for Chebyshev-1 Basis with $\delta_1 = 0$ when the sparsity is $2t-1$). Luckily, every evaluation $g(\zeta)$ at $\zeta \in \mathsf{K}$, $\zeta \neq \pm 1$, yields a second evaluation $g(1/\zeta) = g(\zeta)$ at $1/\zeta$ for (6,8) and a second evaluation $g(1/\zeta) = -g(\zeta)$ at $1/\zeta$ for (7,9). An exception is $\zeta = \pm 1$, which is a Prony point, and the algorithm in [1, Sec. 4.1] for Chebyshev-1 Basis used one additional evaluation. Here we show that the extra evaluation can be avoided by exploiting additional structure in the arising Prony problem for $g(y)$, thus achieving the optimal number of evaluations for the new substitution method in all cases; see Section 4. The conversion to ω also allows for a discrete logarithm-based computation of all δ_j, even from values of $T_{\delta_j}(\beta)$ as in the original Lakshman-Saunders algorithm; see [11].

Finally, we consider bases given by the recurrence

$$V_0^{[u,v,w]}(x)=1,\ V_1^{[u,v,w]}(x)=ux+w,$$
$$V_n^{[u,v,w]}(x)=vxV_{n-1}^{[u,v,w]}(x)-V_{n-2}^{[u,v,w]}(x) \text{ for } n\geq 2, \quad (10)$$

where $u, v, w \in \mathsf{K}$, $u \neq 0$, $v \neq 0$ and K is a field. Our algorithm here for the Chebyshev-2 basis generalizes and computes the sparse representation with terms from (10); see Section 5. One may also seek for a given polynomial $f \in \mathsf{K}[x]$ those parameters u, v, w which yield the maximum sparsity for the corresponding basis. We show how to compute in polynomial time in $\deg(f)$ the optimal pairs u, v, w; see Section 6. The problem is analogous to computing the sparsest shift [8] in the standard powers of variables basis.

2. CHEBYSHEV-1 BASIS WITH SPARSITY KNOWN ON INPUT

Let K be a field of $\mathrm{char}(\mathsf{K}) \neq 2$. A black box polynomial $f(x) \in \mathsf{K}[x]$ can be written as a t-sparse linear combination of Chebyshev-1 Polynomials $c_1T_{\delta_1}(x)+\cdots+c_tT_{\delta_t}(x)$, see (5). We seek to determine the coefficients $c_j \in \mathsf{K} \setminus \{0\}$ and the term degrees $\delta_j \in \mathbb{Z}_{\geq 0}$ from evaluations $a_i = f(\,(\omega^i + \omega^{-i})/2\,)$, $(i = 0, 1, \ldots, 2t-1)$ of the black box for $f(x)$, where $\omega \in \mathsf{K}$, $\omega \neq 0$. The term values ω^{δ_j} of the base point ω are required to be sufficiently distinct, and the δ_j to be recoverable from them. We first assume that we know the sparsity t on input. We also assume that we have a factorization algorithm over K and can compute integral δ from ω^δ.

We define

$$g(y) \stackrel{\text{def}}{=} f(\tfrac{y+y^{-1}}{2}) = \textstyle\sum_{j=1}^{t} \frac{c_j}{2}(y^{\delta_j}+y^{-\delta_j}) \in \mathsf{K}[y, y^{-1}]. \quad (11)$$

The function $g(y)$ is a Laurent polynomial. Let $\omega \in \mathsf{K} \setminus \{0\}$ such that for $i \in \{0, 1, \ldots, 2t-1\}$,

$$a_i \stackrel{\text{def}}{=} g(\omega^i) = f(\tfrac{\omega^i+\omega^{-i}}{2}) = f(T_i(\beta)), \beta \stackrel{\text{def}}{=} \tfrac{\omega+\omega^{-1}}{2}, \quad (12)$$

and for $1\leq i_1<i_2\leq t$, $T_{\delta_{i_1}}(\beta)\neq T_{\delta_{i_2}}(\beta)$ if $\delta_{i_1}\neq\delta_{i_2}$. Note that $a_{-i}=a_i$.

LEMMA 2.1. *If $T_{\delta_{i_1}}(\beta) \neq T_{\delta_{i_2}}(\beta)$ for $1 \leq i_1 < i_2 \leq t$, then $\omega^{\delta_{i_1}} \neq \omega^{\delta_{i_2}}$ or $\omega^{\delta_{i_1}} \neq \omega^{-\delta_{i_2}}$.*

PROOF. If $T_{\delta_{i_1}}(\beta) \neq T_{\delta_{i_2}}(\beta)$, then $\omega^{\delta_{i_1}} + \omega^{-\delta_{i_1}} \neq \omega^{\delta_{i_2}} + \omega^{-\delta_{i_2}}$. Hence $(\omega^{\delta_{i_1}}\omega^{\delta_{i_2}} - 1)(\omega^{\delta_{i_1}} - \omega^{\delta_{i_2}}) \neq 0$ and so $\omega^{\delta_{i_1}} \neq \omega^{-\delta_{i_2}}$ or $\omega^{\delta_{i_1}} \neq \omega^{\delta_{i_2}}$. □

LEMMA 2.2. *Let $1 \leq i_1 < i_2 \leq t$. If the set $\{\omega^{\delta_{i_1}}, \omega^{\delta_{i_2}}, \omega^{-\delta_{i_1}}, \omega^{-\delta_{i_2}}\}$ has at least three elements, then $T_{\delta_{i_1}}(\beta) \neq T_{\delta_{i_2}}(\beta)$.*

PROOF. If the set $\{\omega^{\delta_{i_1}}, \omega^{\delta_{i_2}}, \omega^{-\delta_{i_1}}, \omega^{-\delta_{i_2}}\}$ has at least three elements, then "$\omega^{\delta_{i_1}} \neq \omega^{\delta_{i_2}}$ and $\omega^{\delta_{i_1}} \neq \omega^{-\delta_{i_2}}$" and "$\omega^{\delta_{i_1}} \neq \omega^{-\delta_{i_1}}$ or $\omega^{\delta_{i_2}} \neq \omega^{-\delta_{i_2}}$". Then $2T_{\delta_{i_1}}(\beta) = (\omega^{\delta_{i_1}} + \omega^{-\delta_{i_1}}) \neq (\omega^{\delta_{i_2}} + \omega^{-\delta_{i_2}}) = 2T_{\delta_{i_2}}(\beta)$. □

COROLLARY 2.3. *If the set $\{\omega^{-\delta_t}, \omega^{-\delta_{t-1}}, \ldots, \omega^{-\delta_1}, \omega^{\delta_1}, \ldots, \omega^{\delta_{t-1}}, \omega^{\delta_t}\}$ has at least $2t-1$ elements, then $T_{\delta_{i_1}}(\beta)\neq T_{\delta_{i_2}}(\beta)$ for $1\leq i_1<i_2\leq t$.*

We can interpolate the Laurent polynomial (11) with Prony's algorithm [23] from its $2t$ evaluations $a_0, \ldots, a_{2t-1}$. We query the black box polynomial $f(x)$ to get these evaluations. Since $a_{-i} = a_i$, we actually have $4t-1$ evaluations of $g(y)$: $a_{-2t+1}, \ldots, a_0, \ldots, a_{2t-1}$.

Let α be a symbol for a_{2t}. If $\delta_1 = 0$, then a value for α is not needed for computing the term locator polynomial $\Lambda(z)$ for $g(y)$. The corresponding $2t \times 2t$ matrix $H = [a_{i+j-(2t-1)}]_{i,j=0}^{2t-1}$ will then

have been identified by the Berlekamp/Massey algorithm as singular. If $2t = \deg(\Lambda(z))$, then the matrix is identified as non-singular, and $\Lambda(z)$ is computed as a linear form $\Lambda_\alpha(z) = \Lambda^{[0]}(z) + \alpha\Lambda^{[1]}(z)$ from the system

$$\begin{bmatrix} a_{-2t+1} & a_{-2t+2} \ldots a_{-t+1} \ldots & a_{-1} & a_0 \\ a_{-2t+2} & a_{-2t+3} \ldots a_{-t+2} \ldots & a_0 & a_1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{-1} & a_0 \ \ldots \ a_{t-2} & \ldots a_{2t-3} & a_{2t-2} \\ a_0 & a_1 \ \ldots \ a_{t-1} & \ldots a_{2t-2} & a_{2t-1} \end{bmatrix} \begin{bmatrix} 1 \\ \lambda_1 \\ \vdots \\ \lambda_{2t-2} \\ \lambda_{2t-1} \end{bmatrix} = - \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{2t-1} \\ \alpha \end{bmatrix}. \quad (13)$$

In this case, the term locator polynomial of (11) is

$$\Lambda(z) = \prod_{j=1}^{t} ((z - \omega^{\delta_j})(z - \omega^{-\delta_j})) = z^{2t} + \lambda_{2t-1} z^{2t-1} + \cdots + \lambda_1 z + 1$$

and it is a reciprocal polynomial, i.e., $\lambda_{2t-j} = \lambda_j$, ($\lambda_{2t} = \lambda_0 = 1$). We show that (in Theorem 2.5 below), if the set $\{\omega^{-\delta_t}, \omega^{-\delta_{t-1}}, \ldots, \omega^{-\delta_1}, \omega^{\delta_1}, \ldots, \omega^{\delta_{t-1}}, \omega^{\delta_t}\}$ has $2t$ elements, then α is uniquely determined by the symmetry conditions of the coefficients of $\Lambda(z)$. Hence, to determine the δ_j, we do not need to evaluate the black box polynomial $f(x)$ at $T_{2t}(\beta)$ to get a_{2t}.

From the symmetry conditions of the coefficients of $\Lambda(z)$ the system (13) collapses to the following system:

$$\bar{H} \cdot \begin{bmatrix} \lambda_t/2 \\ \lambda_{t-1} \\ \vdots \\ \lambda_1 \end{bmatrix} = - \begin{bmatrix} a_1 + a_{2t-1} \\ a_2 + a_{2t-2} \\ \vdots \\ 2a_t \end{bmatrix} \quad (14)$$

where $\bar{H}$ is a "fold" of the coefficient matrix of (13):

$$\bar{H} = \begin{bmatrix} 2a_{t-1} & \cdots & a_1 + a_{2t-3} & a_0 + a_{2t-2} \\ 2a_{t-2} & \cdots & a_0 + a_{2t-4} & a_1 + a_{2t-3} \\ \vdots & & \vdots & \vdots \\ 2a_0 & \cdots & 2a_{t-2} & 2a_{t-1} \end{bmatrix}. \quad (15)$$

We have that $\bar{H}$ is non-singular:

Lemma 2.4. *The matrix $\bar{H}$ in (15) is non-singular.*

Proof. $\bar{H} = J_t \cdot A$ where A is the non-singular matrix in Lemma 3.2 in [1] (for $r = 0$ and $s = 1$) and $J_t \in \mathsf{K}^{t \times t}$ is the exchange matrix (row-reversed identity matrix). Note that by our assumptions on ω the $T_{\delta_j}(\beta)$ are distinct. □

Therefore, we can determine the coefficients $\lambda_1 = \lambda_{2t-1}, \ldots, \lambda_{t-1} = \lambda_{t+1}$, and λ_t of the term locator polynomial $\Lambda(z)$ by solving the folded system (14).

Theorem 2.5. *If the set $\{\omega^{-\delta_t}, \ldots, \omega^{-\delta_1}, \omega^{\delta_1}, \ldots, \omega^{\delta_t}\}$ has $2t$ elements, then α is uniquely determined by the symmetry conditions of the coefficients of the term locator polynomial $\Lambda(z)$ of (11).*

Proof. If there were two values for α, then the folded system (14) of the system (13) would have two different solutions. Since $\bar{H}$ is non-singular, this is impossible. Hence α is unique. □

Therefore, to compute the term locator polynomial $\Lambda(z)$ of (11) we need $2t$ evaluations: $a_0, \ldots, a_{2t-1}$. A root ρ of the term locator polynomial is of the form $\rho = \omega^{\pm\delta_j}$. We can compute the δ_j in (5) from the (possibly discrete) logarithms of the roots of the term locator polynomial as it is commonly done in [2, 7, 10, 17, 19]. After determining the δ_j, ($j = 1, \ldots, t$), we compute the coefficients c_j in (5) by solving the non-singular transposed Vandermonde system

$$\begin{bmatrix} 1 & \ldots & 1 & 1 & \ldots & 1 \\ \rho_1 & \ldots & \rho_t & \rho_t^{-1} & \ldots & \rho_1^{-1} \\ \vdots & & \vdots & \vdots & & \vdots \\ \rho_1^{2t-1} & \ldots & \rho_t^{2t-1} & \rho_t^{-2t+1} & \ldots & \rho_1^{-2t+1} \end{bmatrix} \cdot \begin{bmatrix} c_1 \\ \vdots \\ c_t \\ c_t \\ \vdots \\ c_1 \end{bmatrix} = \begin{bmatrix} 2a_0 \\ 2a_1 \\ \vdots \\ 2a_{2t-1} \end{bmatrix}. \quad (16)$$

Here the ρ_j and ρ_j^{-1}, ($j = 1, \ldots, t$), are the roots of the term locator polynomial $\Lambda(z)$. An $t^{1+o(1)}$-time algorithm is in [18, Sec. 5].

Remark 2.1. If the set $\{\omega^{-\delta_t}, \omega^{-\delta_{t-1}}, \ldots, \omega^{-\delta_1}, \omega^{\delta_1}, \ldots, \omega^{\delta_{t-1}}, \omega^{\delta_t}\}$ has $2t-1$ elements, then we can determine the coefficients of the term locator polynomial by solving the system

$$\begin{bmatrix} a_{-2t+2} & \cdots & a_{-t+1} & \cdots & a_0 \\ a_{-2t+3} & \cdots & a_{-t+2} & \cdots & a_1 \\ \vdots & & \vdots & & \vdots \\ a_0 & \cdots & a_t & \cdots & a_{2t-2} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ \lambda_1 \\ \vdots \\ \lambda_{2t-2} \end{bmatrix} = - \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{2t-1} \end{bmatrix}.$$

In this case, for only one δ_j, we have $\omega^{\delta_j} = \omega^{-\delta_j}$. After determining the roots of the term locator polynomial, we can identify that specific δ_j. □

To summarize, we collect the steps of our algorithm as follows:

2.1. Algorithm Sparse Chebyshev-1 Interpolation

Input:
- A black box polynomial $f(x) \in \mathsf{K}[x]$ where K is a field with $\mathrm{char}(\mathsf{K}) \neq 2$.
- The sparsity t of $f(x)$.
- $\omega \in \mathsf{K} \setminus \{0\}$ such that the set of term values $\{\omega^{-\delta_t}, \omega^{-\delta_{t-1}}, \ldots, \omega^{-\delta_1}, \omega^{\delta_1}, \ldots, \omega^{\delta_{t-1}}, \omega^{\delta_t}\}$ has at least $2t$ elements or $2t-1$ elements with $\delta_1 = 0$.
- A factorization algorithm over K.
- A integer-valued-logarithm-base-ω algorithm in K.

Output:
- The coefficients c_j and the term degrees δ_j such that $f(x) = \sum_{j=1}^{t} c_j T_{\delta_j}(x) \in \mathsf{K}[x]$.

1. *For $0 \leq i \leq 2t-1$, get the evaluations $a_i = g(\omega^i)$, as in (12), of the Laurent polynomial (11).*
2. *Solve (13) by the Berlekamp/Massey Algorithm to get the coefficients of the term locator polynomial $\Lambda(z)$. Use the symmetry of the term locator polynomial to find the unique α.*
3. *Find all roots of the term locator polynomial. The roots are of the form $\omega^{\pm\delta_j}$. Compute the δ_j from integer logarithms (discrete logarithms if K is a finite field) of the roots of the term locator polynomial.*
4. *Solve the system (16) to get the coefficients c_j.*
5. *Return the δ_j and the c_j.*

3. CHEBYSHEV-2 BASIS WITH SPARSITY KNOWN ON INPUT

Let K be a field of $\mathrm{char}(\mathsf{K}) \neq 2$. We now consider the representation of a black box polynomial $f(x) \in \mathsf{K}[x]$ as a t-sparse linear combination of Chebyshev-2 Polynomials, i.e.,

$$f(x) = c_1 U_{\delta_1}(x) + \cdots + c_t U_{\delta_t}(x) \in \mathsf{K}[x] \quad (17)$$

where $c_j \in \mathsf{K} \setminus \{0\}$ and $\delta_j \in \mathbb{Z}_{\geq 0}$ such that $\delta_1 < \cdots < \delta_t$. Again, we seek to compute the coefficients $c_j \in \mathsf{K} \setminus \{0\}$ and the term degrees $\delta_j \in \mathbb{Z}_{\geq 0}$ from evaluations $f((\omega^i + \omega^{-i})/2)$, ($i = 0, 1, \ldots, 2t-1$) of the black box for $f(x)$, where $\omega \in \mathsf{K}$, $\omega \neq 0$. The term values ω^{δ_j} of the base point ω are required to be sufficiently distinct, and the δ_j to be recoverable from them. Again, we assume that we know the sparsity t.

Our algorithm proceeds as the Chebyshev-1 Algorithm in Section 2 with the following changes: we define

$$g(y) \stackrel{\text{def}}{=} (y - y^{-1})\, f(\tfrac{y+y^{-1}}{2}) = \sum_{j=1}^{t} \tfrac{c_j}{2} (y^{\delta_j+1} - y^{-(\delta_j+1)}) \in \mathsf{K}[y, y^{-1}] \quad (18)$$

(see (7)). The function $g(y)$ is a Laurent polynomial. We Prony interpolate for a base point $\omega \in \mathsf{K} \setminus \{0\}$ the values

$$a_0 = 0, a_i = -a_{-i} \stackrel{\text{def}}{=} g(\omega^i) = (\omega - \omega^{-1})\, f(\tfrac{\omega^i + \omega^{-i}}{2}), \quad (19)$$

$i \in \{1, \ldots, 2t\}$, and assume that the set of shifted term values $\{\omega^{-(\delta_t+1)}, \omega^{-(\delta_{t-1}+1)}, \ldots, \omega^{-(\delta_1+1)}, \omega^{\delta_1+1}, \ldots, \omega^{\delta_{t-1}+1}, \omega^{\delta_t+1}\}$ has $2t$ or $2t-1$ elements. The difficulties which arose in the Chebyshev-1 case due to $i = 0$ yielding a single evaluation do not occur: $a_0 = 0$ needs no evaluation, and we compute the term locator polynomial from $2t$ black box probes. Algorithm 5.1 below is a generalized variant, which uses randomization to compute t.

4. DETERMINISTIC EARLY TERMINATION WITH A SPARSITY BOUND

We now relax the assumption that on input one has the exact sparsity t, but assume that on input one has an upper bound $B \geq t$ for the sparsity. Our objective is to interpolate with exactly $t + B$ evaluations. Here we assume that the black box for f can be queried as our algorithm proceeds.

Let $a_i = g(\omega^i)$, as in (12), where $g(y)$ is given in (11), and

$$\mathcal{H} = \begin{bmatrix} a_{-2t-2B+1} & \cdots & a_{-2} & a_{-1} & a_0 \\ a_{-2t-2B+2} & \cdots & a_{-1} & a_0 & a_1 \\ a_{-2t-2B+3} & \cdots & a_0 & a_1 & a_2 \\ \vdots & & \vdots & \vdots & \vdots \\ a_0 & \cdots & a_{2t+2B-3} & a_{2t+2B-2} & a_{2t+2B-1} \end{bmatrix} \quad (20)$$

where $B \geq t$. We will consider non-singular square submatrices of $\mathcal{H}$ in the right upper corner.

Remark 4.1. *The 0×0 matrix is called the* empty matrix. *The empty matrix is considered to be non-singular. In Algorithm 4.1 Step 2b below, if $r = 0$, then $\mathcal{H}_R$ in Step 3 is the empty matrix. In this case $\Lambda(z) = z^0 = 1$.*

4.1. Algorithm Chebyshev-1 Term Locator Polynomial

Input:
- A black box polynomial $f(x) \in \mathsf{K}[x]$ where K is a field with $\mathrm{char}(\mathsf{K}) \neq 2$.
- An upper bound $B \geq t$ for the sparsity t of $f(x)$.
- $\omega \in \mathsf{K} \setminus \{0\}$ such that the set of term values $\{\omega^{-\delta_t}, \omega^{-\delta_{t-1}}, \ldots, \omega^{-\delta_1}, \omega^{\delta_1}, \ldots, \omega^{\delta_{t-1}}, \omega^{\delta_t}\}$ (21) has $2t$ elements or $2t-1$ elements with $\delta_1 = 0$.

Output: Sparsity t and the term locator polynomial $\Lambda(z)$.

1. *Get the evaluations $a_0, \ldots, a_{B-1}$. If $a_0 = \cdots = a_{B-1} = 0$, then return $t = 0$ and $\Lambda(z) = 1$. Otherwise, proceed to the next step.*
   Here $a_i = g(w^i)$, as in (12), where $g(y)$ is given in (11). Note that $a_{-i} = a_i$. In Lemma 4.1 below, we prove that $a_0 = \cdots = a_{B-1} = 0$ implies $f(x)$ is identically zero, so $t = 0$ and $\Lambda(z) = z^0 = 1$.
2. *Locate a non-singular leading principal submatrix $\mathcal{H}_{2r-1}$ or $\mathcal{H}_{2r}$ of (20) as follows:*

2a. $r_{old} \leftarrow 0$.

2b. *For r from $r_{old} + 1$ to B do the following:*

2(b)i. *Construct* $\mathcal{H}_{2r-1} = \begin{bmatrix} a_{-2r+2} & \cdots & a_0 \\ \vdots & & \vdots \\ a_0 & \cdots & a_{2r-2} \end{bmatrix}$.

If $\mathcal{H}_{2r-1}$ is non-singular, then $r_{old} \leftarrow r$ and break the loop. Otherwise, proceed to the next step.

2(b)ii. *Construct* $\mathcal{H}_{2r} = \begin{bmatrix} a_{-2r+1} & \cdots & a_0 \\ \vdots & & \vdots \\ a_0 & \cdots & a_{2r-1} \end{bmatrix}$.

If $\mathcal{H}_{2r}$ is non-singular, then $r_{old} \leftarrow r$ and break the loop. Otherwise, proceed to the next step.

If there is no such non-singular leading principal submatrix, $\mathcal{H}_{2r-1}$ or $\mathcal{H}_{2r}$, then the given bound B is not correct. Note that, if the term values (21) collapse, this algorithm can return a wrong sparsity and a wrong term locator polynomial.

3. *Let $\mathcal{H}_R$ denote the non-singular matrix constructed at the previous step.*

3a. *If the rank of $\mathcal{H}_R$ is odd (the case $R = 2r-1$), then do the following:*

3(a)i. *Solve*

$$\mathcal{H}_R \cdot \begin{bmatrix} \lambda_0 \\ \vdots \\ \lambda_{2r-2} \end{bmatrix} = - \begin{bmatrix} a_1 \\ \vdots \\ a_{2r-1} \end{bmatrix}.$$

to compute the linear generator $z^{2r-1} + \lambda_{2r-2} z^{2r-2} + \cdots + \lambda_0$.

3(a)ii. *For i from 1 to $B - r$ do the following:*

–*If $\sum_{k=0}^{2r-2} \lambda_k a_{k-1+i} \neq -a_{2r-2+i}$ then go to Step 2b to locate the next non-singular leading principal submatrix. Otherwise, proceed to the next step.*

3(a)iii. *For i from 1 to $B - r + 1$ do the following:*

–*If $\sum_{k=0}^{2r-2} \lambda_k a_{k-2r+3-i} \neq -a_{2-i}$ then go to Step 2b to locate the next non-singular leading principal submatrix $\mathcal{H}_R$. Otherwise, proceed to the next step.*

At this point, we have found no discrepancies. We prove in Theorem 4.2 below that, in this situation, $t = r$.

3(a)iv. *Return $t = r$ and $\Lambda(z) = z^{2r-1} + \lambda_{2r-2} z^{2r-2} + \cdots + \lambda_0$.*

3b. *If the rank of $\mathcal{H}_R$ is even (the case R=2r), then do the following:*

3(b)i. *Compute the linear generator $z^{2r} + \lambda_{2r-1} z^{2r-1} + \cdots + \lambda_0$ of $a_{-2r+1}, \ldots, a_0, \ldots, a_{2r-1}$ as explained in Section 2.*

3(b)ii. *If $\lambda_{2r-1} = \lambda_1, \ldots, \lambda_{r+1} = \lambda_{r-1}$, then proceed to the next step. Otherwise, go to Step 2b to locate the next non-singular leading principal submatrix $\mathcal{H}_R$.*

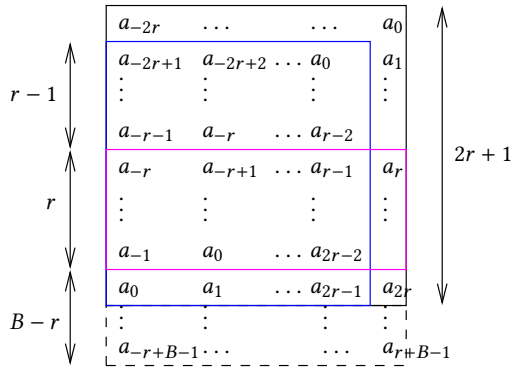
3(b)iii. *For i from 1 to $B - r$ do the following:*

–(Figure 1) *If $\sum_{k=0}^{2r-1} \lambda_k a_{k-1+i} \neq -a_{2r-1+i}$ then go to Step 2b to locate the next non-singular leading principal submatrix $\mathcal{H}_R$. Otherwise, proceed to the next step.*

At this point, we have found no discrepancies. We prove in Theorem 4.2 below that, in this situation, $t = r$.

3(b)iv. *Return $t = r$ and $\Lambda(z) = z^{2r} + \lambda_{2r-1} z^{2r-1} + \cdots + \lambda_0$.*

Figure 1: Intermediate step in Algorithm 4.1



Lemma 4.1. *In Algorithm 4.1 Step 1, if $a_0 = \cdots = a_{B-1} = 0$, then $f(x) = \sum_{j=1}^{t} c_j T_{\delta_j}(x) \in \mathsf{K}[x]$ is identically zero.*

Proof. Let $a_0 = \cdots = a_{B-1} = 0$. Assume that $f(x)$ is a t-sparse non-zero polynomial in the Chebyshev Basis of the first kind. Since $a_i = a_{-i}$ (from (11)), we have $a_{-(B-1)} = \cdots = a_0 = \cdots = a_{B-1} = 0$. Let $\Lambda(z)$ be the term locator polynomial of the Laurent polynomial $g(y)$ (11). The roots $\rho_1, \ldots, \rho_{2t}$ of $\Lambda(z)$ are of the form $\omega^{\pm\delta_j}$, $(1 \leq j \leq t)$. Let $\rho_1 = \omega^{-\delta_t}, \ldots, \rho_t = \omega^{-\delta_1}, \rho_{t+1} = \omega^{\delta_1}, \ldots, \rho_{2t} = \omega^{\delta_t}$. From (11) we know $g(y) = f(\frac{y+y^{-1}}{2}) = \sum_{j=1}^{t} \frac{c_j}{2}(y^{\delta_j} + y^{-\delta_j})$. We can find the coefficients $c_1, \ldots, c_t$ of the Laurent Polynomial $g(y)$ by solving the following system (cf. (16)):
$[\rho_j^i]_{-(B-1) \leq i \leq B-1, 1 \leq j \leq 2t} \cdot [c_t \ldots c_1 c_1 \ldots c_t]^{\mathrm{Tr}} = [2a_i]_{-(B-1) \leq i \leq B-1} = 0$.
From the symmetry conditions of the coefficients of $g(y)$ the above system folds to $\mathcal{R} \cdot [c_1 \ldots c_t]^{\mathrm{Tr}} = [0 \ldots 0]^{\mathrm{Tr}}$ where

$$\mathcal{R} = [\rho_j^{-i} + \rho_j^i]_{i=0,1,\ldots B-1,\ j=t,t-1,\ldots,1}.$$

When $B = t$, the determinant of $\mathcal{R}$ factors as
$\det(\mathcal{R}) = 2/(\rho_1^{t-1}\cdots\rho_t^{t-1}) \cdot \prod_{1\le j<\ell\le t}\Big((\rho_j-\rho_\ell)\cdot(\rho_j\rho_\ell-1)\Big)$,
which is $\neq 0$ because $\rho_j \neq \rho_\ell$ and $\rho_j \neq \rho_\ell^{-1}$ for $1 \le j < \ell \le t$. Therefore, $c_1 = \cdots = c_t = 0$, contradicting to our assumption. Hence, $f(x)$ is identically zero. □

Theorem 4.2. *For the largest non-singular matrix $\mathcal{H}_R$ (where $R = 2r$ or $R = 2r-1$) in Algorithm 4.1, $t = r$.*

Proof. • Case $R = 2r$: Let $\mathcal{H}_R$ be the largest non-singular matrix in Algorithm 4.1. So, $\mathcal{H}_R$ satisfies the condition in Step 3(b)iii.

– Case $\delta_1 > 0$: If $r < t$, then $\mathcal{H}_R$ would not be the last non-singular matrix in Algorithm 4.1 because $\mathcal{H}_{2t}$ is non-singular. If $r > t$, then the folded matrix $\bar{\mathcal{H}}_R$ (which is very similar to (15)) of $\mathcal{H}_R$, which is needed to compute the linear generator in Step 3(b)i in Section 2, would not be non-singular. So $r = t$.
– Case $\delta_1 = 0$: In this case $\mathcal{H}_R$ would be identified as singular. In this situation $\mathcal{H}_{2r-1}$ might be non-singular. This is the next item in the proof.

• Case $R = 2r-1$: Let $\mathcal{H}_R$ be the largest non-singular matrix in Algorithm 4.1. So, $\mathcal{H}_R$ satisfies the conditions in Steps 3(a)ii and Step 3(a)iii. If $r < t$, then $\mathcal{H}_R$ would not be the last non-singular matrix in Algorithm 4.1 because $\mathcal{H}_{2t-1}$ is non-singular. If $r > t$, then the conditions Step 3(a)ii and Step 3(a)iii would push the sparsity t beyond the known bound B. So $r = t$. □

Theorem 4.3. *Algorithm 4.1 requires $t + B$ evaluations.*

Proof. In Step 1, Algorithm 4.1 looks at B evaluations, namely $a_0, \ldots, a_{B-1}$ (note that $a_{-i} = a_i$). Let $\mathcal{H}_R$ (where $R = 2r$ or $R = 2r-1$) be the non-singular matrix constructed in Algorithm 4.1 in Step 2. If $R = 2r$, the algorithm uses $2r$ evaluations in Step 3(b)i (in Step 3(a)i when $R = 2r-1$), namely $a_0, \ldots, a_{2r-1}$. In order to check the linear dependency, it uses $B - r$ evaluations more in Step 3(b)iii (in Step 3(a)iii when $R = 2r-1$), namely $a_{2r}, \ldots, a_{2r-1+(B-r)}$ ($a_{-2r}, \ldots, a_{2-(B-r+1)}$ when $R = 2r-1$). So, the total number of evaluations is $2r + (B-r) = r + B$. Since Algorithm 4.1 terminates when $r = t$, it requires $t + B$ evaluations. □

A difficulty in implementing the algorithm with structured Toeplitz solvers poses Step 2b. By discovering a discrepancy in the column dependency in Steps 3(a)ii, 3(a)iii or 3(b)iii the rank of the $2B\times 2B$ Toeplitz matrix is certified to be larger than the degree R of the current candidate for the term locator polynomial. However, unlike in the Berlekamp-Massey algorithm for Hankel matrices, the dimensions of the new non-singular submatrix can lie beyond the point of the discrepancy. One locates the new non-singular matrix by incremental row elimination of the Schur complements [6, 24], which introduces a running time that is cubic in the distance to the next non-singular Toeplitz submatrix. Alternatively, one could in soft-linear Monte-Carlo time compute the rank of each intermediate Schur complement [15], which yields the $(t+B)^{2+o(1)}$ running time bound cited in the introduction. Note that our matrices can be used to construct symmetric Toeplitz matrices with rational entries that have arbitrary lookahead: for example, the Toeplitz matrix whose first row and first column contain the entries $g_2(2), g_2(2^2), \ldots, g_2(2^{11})$ and whose leading principal submatrices have ranks $1, 2, 2, 2, 2, 2, 4, 6, 8, 10, 11, 11, \ldots$ Here $g_2(x)$ is the symmetric Laurent polynomial $g_2(x) = \frac{32768}{5281339833}(\frac{1}{x^6}+x^6) - \frac{1024}{2540327}(\frac{1}{x^5}+x^5) + \frac{64}{7227}(\frac{1}{x^4}+x^4) - \frac{744}{8687}(\frac{1}{x^3}+x^3) + \frac{62}{153}(\frac{1}{x^2}+x^2) + \frac{254}{189}$. To create that symmetric Toeplitz matrix we started with $g_1(x) = x^{-1}+x$ and then constructed $g_2(x)$ (first with unknown coefficients). Note that $g_1(2^i) = g_2(2^i)$ for $0 \le i \le 5$. A worst-case quadratic-time Toeplitz solver that in analogy to the Berlekamp-Massey Hankel solver incrementally steps from non-singular to non-singular leading principal submatrix is not known to us.

Here we would like to mention about our work in progress [11]. In [11], we give an algorithm for computing the Chebyshev term degrees in the original algorithm of Lakshman and Saunders [20] for a very large finite coefficient field $\mathbb{F}_p$; with a method similar to the Silver-Pohlig-Hellman Algorithm [21], one can directly compute the Chebyshev term degree δ from given $\zeta = T_\delta(\beta)$, $\beta = (\omega + 1/\omega)/2 \in \mathbb{F}_p$, without precomputing the order of $\omega \in \mathbb{F}_p, \omega \neq 0$. In [11], we also show that the same strategy applies to the Silver-Pohlig-Hellman [21] discrete logarithm algorithm to compute δ from given $\zeta = \omega^\delta$; one does not need to precompute the order of ω.

The Chebyshev-2 Basis, Chebyshev-3 Basis, and Chebyshev-4 Basis cases can be done in the same way by making use of the properties (7), (8), and (9). Note that in Chebyshev-2 Basis and Chebyshev-4 Basis cases we have a free evaluation: $a_0 = 0$.

5. SPARSE INTERPOLATION WITH PARAMETERIZED RECURSIVE BASES

We now focus on sparse interpolation in more general polynomial bases, which are defined by the recurrence relation (10), namely

$$V_0^{[u,v,w]}(x) = 1, \quad V_1^{[u,v,w]}(x) = u\,x + w,$$
$$V_n^{[u,v,w]}(x) = v\,x\,V_{n-1}^{[u,v,w]}(x) - V_{n-2}^{[u,v,w]}(x) \text{ for } n \ge 2, \quad (22)$$

where $u, v \in \mathsf{K}\setminus\{0\}$, $w \in \mathsf{K}$ and K is a field. Obviously, Chebyshev-1 through Chebyshev-4 bases are special cases of the above polynomial recurrence bases (22), e.g., $T_n(x) = V_n^{[1,2]}(x) \stackrel{\text{def}}{=} V_n^{[1,2,0]}(x)$, $U_n(x) = V_n^{[2,2]} \stackrel{\text{def}}{=} V_n^{[2,2,0]}(x)$. Note our notation: from now, we omit to write a $w = 0$ in the bracketed superscript. Furthermore, Fact 1.1 can be generalized to the case of the above recurrence bases (22).

Fact 5.1. *Let $u, v \in \mathsf{K}\setminus\{0\}$, $w \in \mathsf{K}$, K is a field, and let $n \in \mathbb{Z}$. Then the following hold:*

i. $\left(x - \frac{1}{x}\right) V_0^{[u,v,w]}\left(\frac{x+\frac{1}{x}}{v}\right) = x - \frac{1}{x}$,
$\left(x - \frac{1}{x}\right) V_1^{[u,v,w]}\left(\frac{x+\frac{1}{x}}{v}\right) = \frac{u}{v}\left(x^2 - \frac{1}{x^2}\right) + w\left(x - \frac{1}{x}\right)$.
ii. $\left(x - \frac{1}{x}\right) V_n^{[u,v,w]}\left(\frac{x+\frac{1}{x}}{v}\right) = \frac{u}{v}\left(x^{n+1} - \frac{1}{x^{n+1}}\right) + w\left(x^n - \frac{1}{x^n}\right) + \left(\frac{u}{v} - 1\right)\left(x^{n-1} - \frac{1}{x^{n-1}}\right)$ *for all $n \in \mathbb{Z}$.*

Remark 5.1. If $u = v \neq 0 \in \mathsf{K}$, $w = 0$, Fact 5.1 implies

$$\left(x - \frac{1}{x}\right) V_n^{[v,v]}\left(\frac{x+\frac{1}{x}}{v}\right) = x^{n+1} - \frac{1}{x^{n+1}} \text{ for all } n \ge 1. \quad (23)$$

From $U_n(x) = V_n^{[2,2]}$ we obtain (7). The binomial solutions (8,9) generalize similarly for $u = v$ and $w = \pm 1$. Furthermore, given a recurrence basis $V_n^{[u,v,w]}(x)$, then for each $\sigma \in \mathsf{K}\setminus\{0\}$ and $n \in \mathbb{Z}$ we have $V_n^{[u,v,w]}(x) = V_n^{[\frac{u}{\sigma}, \frac{v}{\sigma}, w]}(\sigma x)$. □

A polynomial $f(x)$ is represented as

$$f(x) = \textstyle\sum_{j=1}^t c_j V_{\delta_j}^{[u,v,w]}(x) \in \mathsf{K}[x], 0 \le \delta_1 < \cdots < \delta_t, \forall j\colon c_j \neq 0. \quad (24)$$

Here we say that $f(x)$ is t-sparse in the recurrence basis (with parameters u, v, w). Suppose a black box of $f(x)$ is given to return the evaluation $f(\omega)$ for any $\omega \in \mathsf{K}$. By performing the substitution in Fact 5.1, we have

$$g(y) = \left(y - \frac{1}{y}\right) f\left(\frac{y+\frac{1}{y}}{v}\right) = \textstyle\sum_{j=1}^t c_j \times \left(\frac{u}{v}\left(y^{\delta_j+1} - \frac{1}{y^{\delta_j+1}}\right) + w\left(y^{\delta_j} - \frac{1}{y^{\delta_j}}\right) + \left(\frac{u}{v} - 1\right)\left(y^{\delta_j-1} - \frac{1}{y^{\delta_j-1}}\right)\right)$$
$$\stackrel{\text{def}}{=} \textstyle\sum_{j=1}^\tau g_j \left(y^{\gamma_j} - \frac{1}{y^{\gamma_j}}\right) \in \mathsf{K}[y, \frac{1}{y}], \quad g_j \neq 0 \text{ for all } j, \quad (25)$$

where $1 \le \gamma_1 < \gamma_2 < \cdots < \gamma_\tau$ and 2τ is the sparsity of the Laurent polynomial g in the power basis with $\tau \le 3t$. By (25) the degrees satisfy

$\gamma_\tau=\delta_t+1$. Note that $f(x)=g(z)/(z-\frac{1}{z})$ for $z=(vx\pm(v^2x^2-4)^{1/2})/2$.

Now we present an algorithm to interpolate $f(x)$ from the evaluations of the form

$$a_i=-a_{-i}=\left(\omega^i-\frac{1}{\omega^i}\right)f\left(\frac{\omega^i+\omega^{-i}}{v}\right)\in\mathsf{K},\ \omega\in\mathsf{K},\ \omega\neq 0,\ i=0,1,2,\ldots \quad (26)$$

Let $\rho_j=\omega^{\gamma_j}$, $j=1,\ldots,\tau$, and define the term locator polynomial $\Lambda(z)$ as

$$\Lambda(z)=\prod_{j=1}^{\tau}(z-\rho_j)\,(z-\tfrac{1}{\rho_j}) = z^{2\tau}+\lambda_{2\tau-1}z^{2\tau-1}+\cdots+\lambda_1 z+\lambda_0\in\mathsf{K}[z]. \quad (27)$$

Note that (27) is a reciprocal polynomial, that is $\lambda_0=1$ and $\lambda_j=\lambda_{2\tau-j}$. Similar to the fact stated before Lemma 4.1 in [1], we have that the sequence of values (26) is linearly generated by the polynomial $\Lambda(z)$, but Λ is the minimal generator only if Λ is squarefree, that is, if the term values are distinct. We can determine τ by early termination as in [1, Section 4.2]. Let

$$\alpha_i=-\alpha_{-i}=g(y^i)=\left(y^i-\frac{1}{y^i}\right)f\left(\frac{y^i+\frac{1}{y^i}}{v}\right)\in\mathsf{K}\left[y,\tfrac{1}{y}\right],\quad i\in\mathbb{Z},$$

be the evaluations at powers of the variable y for the ω^i. For the evaluations α_i, $-2\tau-1\le i\le 2\tau+1$, we consider the square Hankel matrix

$$\mathcal{H}=\begin{bmatrix}\alpha_{-2\tau-1}&\alpha_{-2\tau}&\cdots&\alpha_{-1}&\alpha_0\\ \alpha_{-2\tau}&\alpha_{-2\tau+1}&\cdots&\alpha_0&\alpha_1\\ \vdots&\vdots&\iddots&\vdots&\vdots\\ \alpha_{-1}&\alpha_0&\cdots&\alpha_{2\tau-1}&\alpha_{2\tau}\\ \alpha_0&\alpha_1&\cdots&\alpha_{2\tau}&\alpha_{2\tau+1}\end{bmatrix} \quad (28)$$

$\in\mathsf{K}\left[y,y^{-1}\right]^{(2\tau+2)\times(2\tau+2)}$. As in [1, Theorem 4.3.i], the square submatrices in the right upper corner have the following guaranteed non-singularities.

Theorem 5.2. *Let $\mathcal{H}_i$ be the submatrix of $\mathcal{H}$ formed by the first i rows and the last i columns. Then $\det(\mathcal{H}_i)\neq 0$ for $i=2,4,\ldots,2\tau$, and $\det(\mathcal{H}_{2\tau+1})=\det(\mathcal{H}_{2\tau+2})=0$, where $\mathcal{H}_{2\tau+2}=\mathcal{H}$ in (28).*

Proof. The proof of Theorem 4.3.i in [1] is for a Laurent polynomial

$$\sum_{j=1}^{\tau}g_j\left(y^{\delta_j}+y^{-\delta_j}\right)\in\mathsf{K}\left[y,y^{-1}\right],\quad g_j\neq 0, \quad (29)$$

which is [1, Eq. (16)] with $\tau=t$ and $g_j=c_j/2$. Part i of that Theorem includes $\det(\mathcal{H}_{2\tau})\neq 0$ for $\delta_1\ge 1$, which is a property of the degrees γ_j of our terms in (25). The coefficients of our terms in $g(y)$ in (25) are negated for negative term degrees, which is the only difference to (29). Since the proof of Theorem 4.3.i does not use any relation between the coefficients other than they being non-zero (the denominator 2 plays the role of v and could be divided into the coefficient), Part i also holds for the polynomial (25) here.

The singularities of $\mathcal{H}_{2\tau+1}$ and $\mathcal{H}$ follow from the fact that the polynomial $\prod_{j=1}^{\tau}(z-y^{\gamma_j})(z-y^{-\gamma_j})$ is a linear generator for the infinite sequence α_i and its coefficients yield a column relation for $2\tau+1$ consecutive columns in $\mathcal{H}$. □

Before recovering $f(x)$ in sparse representation in the recurrence basis, we present an early termination algorithm to interpolate the Laurent polynomial $g(y)=(y-1/y)\,f((y+1/y)/v)$ in (25) from the univariate black box polynomial $f(x)$. Suppose ω is selected randomly and uniformly from a sufficiently large finite set of field elements $S\subseteq\mathsf{K}\setminus\{0\}$. For $k=1,2,3,\ldots$ we compute the two new values $a_i=(\omega^i-\omega^{-i})\,f((\omega^i+\omega^{-i})/v)$, $i=2k-2,2k-1$, and the determinants of the $(2k)\times(2k)$ Hankel matrices which with

$$H_{2k}=\begin{bmatrix}-a_{2k-1}&-a_{2k-2}&\cdots&-a_1&a_0\\ -a_{2k-2}&-a_{2k-3}&\cdots&a_0&a_1\\ \vdots&\vdots&\iddots&\vdots&\vdots\\ -a_1&a_0&\cdots&a_{2k-3}&a_{2k-2}\\ a_0&a_1&\cdots&a_{2k-2}&a_{2k-1}\end{bmatrix}, \quad (30)$$

$a_{-i}=-a_i$ are the determinants of $\mathcal{H}_{2k}$ in Theorem 5.2 for the evaluation $y=\omega$.

We terminate the loop when $\det(H_{2k})=0$, which implies that the number of terms in $g(y)$ is $2k-2$ with high probability, i.e., $\tau=k-1$. Suppose now that $k-1=\tau$. Then we get the minimal linear generator $\Lambda(z)$ in (27) by solving the following non-singular linear system:

$$\underbrace{\begin{bmatrix}-a_{2\tau-1}&-a_{2\tau-2}&\cdots&-a_1&a_0\\ -a_{2\tau-2}&-a_{2\tau-3}&\cdots&a_0&a_1\\ \vdots&\vdots&\iddots&\vdots&\vdots\\ -a_1&a_0&\cdots&a_{2\tau-3}&a_{2\tau-2}\\ a_0&a_1&\cdots&a_{2\tau-2}&a_{2\tau-1}\end{bmatrix}}_{H_{2\tau}}\cdot\begin{bmatrix}\lambda_0\\ \lambda_1\\ \vdots\\ \lambda_{2\tau-2}\\ \lambda_{2\tau-1}\end{bmatrix}=-\begin{bmatrix}a_1\\ a_2\\ \vdots\\ a_{2\tau-1}\\ a_{2\tau}\end{bmatrix}. \quad (31)$$

Note that because $\det(H_{2\tau})\neq 0$ implies $\Lambda(z)$ in (27) must be squarefree (cf. Lemma 4.2 in [1]), and with $\lambda_1=\lambda_{2\tau-1}$, $\lambda_2=\lambda_{2\tau-2},\ldots$ the system (31) is overdetermined.

Next, we compute all 2τ distinct roots of $\Lambda(z)$, which are ω^{γ_j} and $\omega^{-\gamma_j}$ for $j=1,\ldots,\tau$. Finally, we compute all the coefficients g_j in (25) by solving a $(2\tau)\times(2\tau)$ non-singular transposed Vandermonde system:

$$\begin{bmatrix}1&\cdots&1&1&\cdots&1\\ \rho_1&\cdots&\rho_\tau&\rho_\tau^{-1}&\cdots&\rho_1^{-1}\\ \vdots&&\vdots&\vdots&&\vdots\\ \rho_1^{2\tau-1}&\cdots&\rho_\tau^{2\tau-1}&\rho_\tau^{-2\tau+1}&\cdots&\rho_1^{-2\tau+1}\end{bmatrix}\cdot\begin{bmatrix}g_1\\ \vdots\\ g_\tau\\ -g_\tau\\ \vdots\\ -g_1\end{bmatrix}=\begin{bmatrix}a_0\\ a_1\\ \vdots\\ a_{2\tau-1}\end{bmatrix}. \quad (32)$$

Again, the system (32) is overdetermined. For the given u,v, the coefficients c_j of $f(x)$ can be obtained by solving a linear system obtained from (25). Given the recurrence basis $V_n^{[u,v,w]}(x)$, for given u,v,w, Algorithm 5.1 below recovers $f(x)=\sum_{j=1}^{t}c_jV_{\delta_j}^{[u,v,w]}(x)$ from the black box.

5.1. Algorithm Sparse Interpolation in a Given Recurrence Basis With Early Termination

Input: ▸ $f(x)\in\mathsf{K}[x]$ input as a black box.
▸ u,v,w: the recursive basis parameters for $V_n^{[u,v,w]}(x)$.

Output: ▸ $f(x)=\sum_{j=1}^{t}c_jV_{\delta_j}^{[u,v,w]}(x)$, where $c_j\neq 0$.

1. *Pick a random element ω from a finite set $S\subseteq\mathsf{K}$.*
2. *Determine the number of terms of $g(y)$.*
   For $i=1,2,3,\ldots$ do
   2a. *Get the evaluations $a_i=(\omega-\frac{1}{\omega})f(\frac{\omega^i+\omega^{-i}}{v})$ from the black box of $f(x)$, and then construct the Hankel matrix H_{2k} from $a_1,\ldots,a_{2k-1}$.*
   2b. *Check whether H_{2k} is singular. If $\det(H_{2k})=0$, and then break out of the loop.*
3. *Find the minimal linear generator $\Lambda(z)$ by solving the system (31).*
4. *Compute the roots ρ_j of $\Lambda(z)$, and recover the exponents γ_j of $g(y)$.*
5. *Obtain the coefficients g_j of $g(y)$ by solving the transposed Vandermonde system (32).*
6. *Compute the coefficients c_j of $f(x)$ from (25).*

6. COMPUTING SPARSEST REPRESENTATION

Given a recursive basis $V_n^{[u,v,w]}(x)$ (22), the representation of $f(x)$ in this given basis $V_n^{[u,v,w]}(x)$ is unique. However, different recursive bases, i.e., different u,v,w might change the sparsities of the corresponding representations. For instance,

$$f(x)=\tfrac{1}{2}V_{99}^{[2,2]}=V_1^{[1,2]}(x)+V_3^{[1,2]}(x)+\cdots+V_{97}^{[1,2]}(x)+V_{99}^{[1,2]}(x). \quad (33)$$

(note that we write $V_n^{[u,v]}\stackrel{\text{def}}{=}V_n^{[u,v,0]}$). Therefore, the sparsity of the representation of $f(x)$ depends on the selected u,v,w of the

recursive basis.

In this section, we focus on the choice of the recursive basis such that the representation of $f(x)$ is sparsest, namely, on how to compute u, v, w such that the number of non-zero terms t is minimized in (24). Given a black box of $f(x)$, we first discuss how to recover the Laurent polynomial $g(y)$ in (25) such that the sparsity is optimized over the control variable $v \neq 0$; since the sparsity of (25) is only dependent on the ratio u/v, one may set $u = 1$. Let $g^{[v]}(y) = (y-1/y)f((y+1/y)/v)$. The sparsity of $g^{[v]}(y)$ is clearly dependent on the choice of v. For example, if we construct the Laurent polynomials $g^{[v]}(y)$ from $f(x) = \frac{1}{2}V_{99}^{[1,2]}(x)$ by selecting two different $v = 1, 2$, that is,

$$g^{[1]}(y) = (y - \tfrac{1}{y})f(y + \tfrac{1}{y}) = \textstyle\sum_{j=1}^{50} g_{2i}\,(y^{2i} - y^{-2i}),\ g_{2i} \neq 0,$$

$$g^{[2]}(y) = (y - \tfrac{1}{y})f(\tfrac{y+\frac{1}{y}}{2}) = \tfrac{1}{2}y^{100} - \tfrac{1}{2}y^{-100}.$$

It is easy to see that $g^{[1]}(y)$ has 100 non-zero terms, whereas $g^{[2]}(y)$ has 2 non-zero terms.

In this paper, we also strive to minimize the number of evaluations to interpolate $f(x)$. To that end, we determine v such that the number of the non-zero terms in $g^{[v,w]}(y)$ is minimized. Giesbrecht, Kaltofen and Lee [8] introduces the fraction-free Berlekamp/Massey algorithm for computing the sparsest shifts of a given polynomial. This method can be easily adapted for tackling the problem of computing v such that $g^{[v]}(y)$ is sparsest. We now describe a probabilistic algorithm, given in [8], for recovering the sparsest Laurent polynomial $g^{[v]}(y)$ by the combination of the fraction-free Berlekamp/Massey algorithm with a GCD procedure. Let v be an indeterminate, and choose distinct random values $p, q \in S \subseteq \mathsf{K}$. At first two sequences α_i and β_i are constructed as following:

$$\alpha_i = g^{[\frac{1}{v}]}(p^i) = (p^i - \tfrac{1}{p^i})f(v\,p^i + \tfrac{v}{p^i}) \in \mathsf{K}[v],$$

$$\beta_i = g^{[\frac{1}{v}]}(q^i) = (q^i - \tfrac{1}{q^i})f(v\,q^i + \tfrac{v}{q^i}) \in \mathsf{K}[v].$$

For $i = 1, 2, \ldots$, the discrepancies $\Delta_i(p) \in \mathsf{K}[v]$ and $\Delta_i(q) \in \mathsf{K}[v]$ are obtained by performing the fraction-free Berlekamp/Massey algorithm on the sequences: α_i and β_i. We terminate the loop when $\Gamma = \gcd(\Delta_i(p), \Delta_i(q))$ has a non-zero root ζ in $\overline{\mathsf{K}}$, the algebraic closure of K. In addition, the fraction-free Berlekamp/Massey algorithm yields the corresponding minimal generators of $(\alpha_i)_{i\geq 0}$ and $(\beta_i)_{i\geq 0}$. In the end, we obtain a sparsest Laurent polynomial $g^{[v^*]}(y)$, with $v^* = 1/\zeta$ by performing Steps 4 and 5 in Algorithm 5.1. The probabilistic analysis can be found in [8].

Given a black box of $f(x)$, the above method can be applied to obtain v^* and the sparsest Laurent polynomial $g^{[v^*]}(y) = (y-1/y) \times f((y+1/y)/v^*)$. The sparseness of $g^{[v^*]}(y)$ is by Fact 5.1.ii no more than 6 times the sparsity for the optimal u, v, w values. Note that by (23) the representation of f in the recurrence basis with with $u = v^*$ and $w = 0$ basis has sparsity twice the sparsity of $g^{[v^*]}(y)$ in standard power basis.

Example 6.1. Consider the polynomial $f(x) = 16\,x^5 - 16\,x^3 + 3\,x$, and two representations of $f(x)$ in two different orthogonal bases:

$$f(x) = R_1(x) = -102\,V_1^{[-\frac{1}{2},1]}(x) - 32\,V_5^{[-\frac{1}{2},1]}(x),$$

$$f(x) = R_2(x) = \tfrac{1}{16}\,V_1^{[4,2]}(x) - \tfrac{1}{8}\,V_3^{[4,2]}(x) + \tfrac{1}{4}\,V_5^{[4,2]}(x).$$

For the bases $V^{[-\frac{1}{2},1]}(x)$ and $V^{[4,2]}(x)$, we can get the corresponding Laurent polynomials:

$$g^{[1]}(y) = \left(y - \tfrac{1}{y}\right) f\left(y + \tfrac{1}{y}\right) = 16\left(y^6 - \tfrac{1}{y^6}\right) + 48\left(y^4 - \tfrac{1}{y^4}\right) + 51\left(y^2 - \tfrac{1}{y^2}\right),$$

$$g^{[2]}(y) = \left(y - \tfrac{1}{y}\right) f\left(\tfrac{y+\frac{1}{y}}{2}\right) = \tfrac{1}{2}\left(y^6 - \tfrac{1}{y^6}\right). \tag{34}$$

One can see that the representation $R_1(x)$ is sparser than the representation $R_2(x)$, even though $g^{[1]}(y)$ has more terms by comparison with $g^{[2]}(y)$. Of course, by (34) we must have $f(x) = U_5(x)/2$. □

We do not know an example of a polynomial f where the sparsities in recurrence bases with parameters u, v^*, w, where $v^* \neq 0$ minimizes the sparsity of $g^{[v^*]}(y)$ (25), are larger for all $u \neq 0$ and w than the minimal sparsity that is achieved by a recurrence basis with parameters u', v', w', $u' \neq 0$, $v' \neq 0$ and $v^* \neq v'$. One may compute optimal $u', v', w' \in \overline{\mathsf{K}}$, where $\overline{\mathsf{K}}$ is the algebraic closure of the field K, in time that is polynomial in $\deg(f)$. The algebraic elements u', v', w' are represented in terms of the roots of a polynomial. One computes the coefficients $c_j(u, v, w)$ in (24) for symbolic u, v, w. Because the leading coefficient of $V_\delta^{[u,v,w]}$ is equal $uv^{\delta-1}$, the denominator of the rational function $c_j(u, v, w)$ is a power-term in u, v. We now seek a point $(u', v', w') \in \overline{\mathsf{K}}^3$ that is a zero of a maximum number of the numerator polynomials of $c_j(u, v, w)$. The arising polynomial root finding problem is solvable in polynomial-time in $\deg(f)$. For example, the 0- and 2-dimensional components that zero a maximal number of coefficients are computed via a GCD-free basis computation [3, 13] of the numerator polynomials. Those common factors that occur most often constitute those components. We will analyze the actual complexity of zeroing a maximum number of polynomials in an inconsistent polynomial system elsewhere. The defining equations for the algebraic extensions can be factored lazily by GCDs rather than polynomial factorization over K (cf. [12]).

Some special cases can be treated by linear algebra. We now present a theorem to show the feasibility of how to select for a given v and $w = 0$ a suitable u in the recurrence basis (22) such that $f(x)$ has the sparsest representation, i.e., how to determine $u \in \overline{\mathsf{K}}, u \neq 0$ for a fixed $v \in \mathsf{K}, v \neq 0$ such that the representation of $f(x)$ in the basis $V_n^{[u,v]}(x) = V_n^{[u,v,0]}(x)$ is the sparsest.

Theorem 6.1. *Let $f(x) = \sum_{j=0}^{d} f_j x^j \in \mathsf{K}[x]$ with $d = \deg(f) \geq 2$, where K is a field, and let $v \in \overline{\mathsf{K}}, v \neq 0$. For i with $0 \leq i \leq d-2$ define*

$$S_i \stackrel{\text{def}}{=} \{u \in \overline{\mathsf{K}} \mid f(x) = \textstyle\sum_{j=0}^{d} c_j V_j^{[u,v]}(x) \text{ with } c_i = 0\}. \tag{35}$$

i. *If $d - i \geq 2$ is even, then $|S_i| \leq (d-i)/2$.*

ii. *If $d - i \geq 3$ is odd and $\exists k, 1 \leq k \leq \lfloor (d-i)/2 \rfloor$: $f_{i+2k} \neq 0$, then $|S_i| \leq \lfloor (d-i)/2 \rfloor$.*

Proof. We first prove Part i. Let $g(y) = (y - \frac{1}{y})f(\frac{y+\frac{1}{y}}{v}) \in \mathsf{K}(y)$. By (25) and we can see that $g(y)$ is of the form

$$g(y) = \textstyle\sum_{j=1}^{d+1} g_j(y^j - y^{-j}), \tag{36}$$

where $g_j \in \mathsf{K}$ for $j = 1, \ldots, d+1$, and $g_{d+1} \neq 0$. Let $u, c_0, \ldots, c_d$ be parameters, and suppose $p(x, u, c_0, \ldots, c_d) = \sum_{j=0}^{d} c_j V_j^{[u,v]}(x)$. We have from Fact 5.1 that

$$(y - \tfrac{1}{y})p(\tfrac{y+\frac{1}{y}}{v}, u, c_0, \ldots, c_d) = c_d\,\tfrac{u}{v}\,(y^{d+1} - y^{-d-1}) +$$

$$c_{d-1}\,\tfrac{u}{v}\,(y^d - y^{-d}) + \left(\textstyle\sum_{j=1}^{d-2}\left(c_j\,\tfrac{u}{v} + c_{j+2}\,(\tfrac{u}{v} - 1)\right)(y^{j+1} - y^{-j-1})\right)$$

$$+ c_2(\tfrac{u}{v} - 1)\,(y - y^{-1}) + c_0(y - y^{-1}). \tag{37}$$

According to the definition (35) of S_i, we need find $u, c_0, \ldots, c_u \in \overline{\mathsf{K}}$ that satisfy $c_i = 0$, and the following equation

$$g(y) - (y - \tfrac{1}{y})\,p(\tfrac{y+\frac{1}{y}}{v}, u, c_0, \ldots, c_d) = 0. \tag{38}$$

Since $c_i = 0$ and $d-i$ is even, here we can get the following equations by selecting the coefficients of (38) corresponding to y^{i+1}, y^{i+3}, $\ldots$, y^{d-1}, y^{d+1}:

$$(\tfrac{u}{v} - 1)c_{i+2} - g_{i+1} = 0,\ (\tfrac{u}{v} - 1)c_{i+4} + \tfrac{u}{v}c_{i+2} - g_{i+3} = 0, \ldots,$$

$$(\tfrac{u}{v} - 1)\,c_d + \tfrac{u}{v}\,c_{d-2} - g_{d-1} = 0,\ \tfrac{u}{v}\,c_d - g_{d+1} = 0,$$

whose matrix form is (39) below. Moreover, the dimension of the matrix in (39) is $(\frac{d-i}{2} + 1) \times \frac{d-i}{2}$.

In the following, two cases will be discussed: $g_{i+1} = 0$ and $g_{i+1} \neq 0$.

$$\begin{bmatrix} \frac{u}{v}-1 & 0 & 0 & \cdots & \cdots & 0 \\ \frac{u}{v} & \frac{u}{v}-1 & 0 & \ddots & \ddots & \vdots \\ 0 & \frac{u}{v} & \frac{u}{v}-1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \frac{u}{v} & \frac{u}{v}-1 & 0 \\ 0 & \cdots & \cdots & 0 & \frac{u}{v} & \frac{u}{v}-1 \\ 0 & 0 & \cdots & \cdots & 0 & \frac{u}{v} \end{bmatrix} \cdot \begin{bmatrix} c_{i+2} \\ c_{i+4} \\ \vdots \\ \vdots \\ c_{d-2} \\ c_d \end{bmatrix} = \begin{bmatrix} g_{i+1} \\ g_{i+3} \\ \vdots \\ \vdots \\ g_{d-1} \\ g_{d+1} \end{bmatrix}. \quad (39)$$

We first consider the first case: $g_{i+1} = 0$. We shall investigate the structure of (39). It is easy to check that the above overdetermined linear system is consistent if $u = v$, that implies $v \in S_i$. Now let us consider $u \neq v$. The above linear system (39), removing the last equation consists of a square bidiagonal linear system, whose unique solution is expressed as $c_{i+2} = 0$, $c_{i+4} = g_{i+3}/(\frac{u}{v}-1)$, and so on. Finally, c_d must be of the form $c_d = q_1(u)/(\frac{u}{v}-1)^l$, where $l = \frac{d-i}{2} - 1$ with $q_1(u) \in \mathsf{K}[u]$ and $\deg(q_1) \leq l-1$. Furthermore c_d must satisfy the last equation in (39), that is,

$$\psi_1(u) \stackrel{\text{def}}{=} g_{d+1}\left(\frac{u}{v}-1\right)^l - \frac{u}{v} q_1(u) = 0. \quad (40)$$

Since $g_{d+1} \neq 0$, $\psi_1(0) \neq 0$ and therefore $\psi_1(u)$ is a nonzero polynomial in $\mathsf{K}[u]$, and $\deg(\psi_1(u)) \leq l$. Therefore, for S_i we have the subset relation

$$S_i \subseteq \{v\} \cup \{\bar{u} \mid \psi_1(\bar{u}) = 0, \psi_1 \in \mathsf{K}[u], \psi_1 \neq 0, \text{ with } \deg(\psi_1(u)) \leq (d-i)/2 - 1\}, \quad (41)$$

which implies that $|S_i| \leq \frac{d-i}{2}$.

Next, we consider the other case: $g_{i+1} \neq 0$. A necessary condition that the linear system (39) is consistent is $u \neq v$, by the first row. Similarly, one can obtain $c_{i+2} = g_{i+1}/\left(\frac{u}{v}-1\right)$, $c_{i+4} = \left((g_{i+3}-g_{i-1})\frac{u}{v} - g_{i+3}\right)/(\frac{u}{v}-1)^2$, and so on. Finally, c_d is of the form $c_d = q_2(u)/(\frac{u}{v}-1)^{l+1}$, where $q_2(u) \in \mathsf{K}[u]$ with $\deg(q_2(u)) \leq l$. By substituting the solution of c_d into the last equation of (39), we have $\psi_2(u) \stackrel{\text{def}}{=} g_{d+1}(\frac{u}{v}-1)^{l+1} - \frac{u}{v} q_2(u) = 0$. Likewise, for S_i we have the subset relation

$$S_i \subseteq \{\bar{u} \mid \psi_2(\bar{u}) = 0, \text{ with } \psi_2(u) \neq 0, \deg(\psi_2(u)) \leq \frac{d-i}{2}\}, \quad (42)$$

which implies that $|S_i| \leq \frac{d-i}{2}$.

To conclude, we prove Part ii. According to our assumption, there exists a j such that $j-i = 2k$ is a positive even integer and $f_j \neq 0$. Among all such j, we select the largest j, denote by e. In other words, for e we have $e < d$, and $f_{d-1} = f_{d-3} = \cdots = f_{e+2} = 0$, $f_e \neq 0$. Therefore, the polynomial $f(x)$ can be expressed as $f(x) = f^{[1]}(x) + f^{[2]}(x)$, with

$$f^{[1]}(x) = \sum_{\kappa=0}^{\lfloor d/2 \rfloor} f_{d-2\kappa} x^{d-2\kappa}, \quad f^{[2]}(x) = \sum_{\kappa=0}^{\lfloor e/2 \rfloor} f_{e-2\kappa} x^{e-2\kappa},$$

and $f_d \neq 0$, $f_e \neq 0$. Define $S_i^{[2]} \stackrel{\text{def}}{=} \{u \in \mathsf{K} \mid f^{[2]}(x) = \sum_{j=0}^{e} \bar{c}_j V_j^{[u,v]}(x)$ with $\bar{c}_i = 0\}$. Since $\deg(f^{[2]}) - i = e - i$ is even, and applying Part i to $f^{[2]}(x)$ indicates that $|S_i^{[2]}| \leq \frac{e-i}{2} \leq \lfloor \frac{d-i}{2} \rfloor$. Let $g^{[1]}(y) = (y - \frac{1}{y}) f^{[1]}(\frac{y+\frac{1}{y}}{v})$ and $g^{[2]}(y) = (y - \frac{1}{y}) f^{[2]}(\frac{y+\frac{1}{y}}{v})$. It can be seen that there are no common terms between $g^{[1]}(y)$ and $g^{[2]}(y)$, which implies that $S_i = S_i^{[2]}$. So $|S_i| \leq \lfloor \frac{d-i}{2} \rfloor$. □

Given a polynomial $f(x) = \sum_{j=0}^{d} f_j x^j$, and i chosen from Part i or Part ii of Theorem 6.1, one is able to compute all $u \in \overline{\mathsf{K}}$ such that $f(x) = \sum_{j=0}^{d} c_j V_j^{[u,v]}(x)$ with $c_i = 0$. The second-highest term coefficient c_{d-1}/constant coefficient c_0 is zero/non-zero if and only if g_d/g_1 in (36) is zero/non-zero, independently of the choice of u, v (see (37)). The minimal polynomials for the candidate algebraic number $\bar{u}$ from (41, 42) need not be factored and lazy factorization in the zero tests of elements in the algebraic extensions can be applied (cf. [12]). For each candidate u one can count the number of zero coefficients in (24) and select those with smallest sparsity.

ACKNOWLEDGMENTS

Supported in part by the National Sci. Found. under Grants CCF-1421128 and 1717100 (Imamoglu and Kaltofen), and by the Chinese National Natural Sci. Found. under Grant 61772203 and the Shanghai Natural Sci. Found. under Grant 17ZR1408300 (Yang).

REFERENCES

[1] Andrew Arnold and Erich L. Kaltofen. 2015. Error-Correcting Sparse Interpolation in the Chebyshev Basis. In *ISSAC'15 Proc. 2015 ACM Internat. Symp. Symbolic Algebraic Comput.* ACM, New York, N. Y., 21–28. URL: EKbib/15/ArKa15.pdf.

[2] M. Ben-Or and P. Tiwari. 1988. A Deterministic Algorithm for Sparse Multivariate Polynomial Interpolation. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing (STOC '88)*. ACM, New York, NY, USA, 301–309. https://doi.org/10.1145/62212.62241

[3] Daniel J. Bernstein. 2005. Factoring into coprimes in essentially linear time. *J. Algorithms* 54, 1 (2005), 1–30. https://doi.org/10.1016/j.jalgor.2004.04.009

[4] R. P. Brent, F. G. Gustavson, and D. Y. Y. Yun. 1980. Fast solution of Toeplitz systems of equations and computation of Padé approximants. *J. Algorithms* 1 (1980), 259–295.

[5] C. Brezinski. 1991. *History of Continued Fractions and Padé Approximants*. Springer Verlag, Heidelberg, Germany.

[6] T. F. Chan and P. C. Hansen. 1992. A look-ahead Levinson algorithm for general Toeplitz systems. *IEEE Transactions on Signal Processing* 40, 5 (May 1992), 1079–1090. https://doi.org/10.1109/78.134471

[7] S. Garg and É. Schost. 2009. Interpolation of polynomials given by straight-line programs. *Theoretical Computer Science* 410, 27-29 (2009), 2659–2662.

[8] Mark Giesbrecht, Erich Kaltofen, and Wen-shin Lee. 2003. Algorithms for Computing Sparsest Shifts of Polynomials in Power, Chebychev, and Pochhammer Bases. *J. Symbolic Comput.* 36, 3–4 (2003), 401–424. URL: EKbib/03/GKL03.pdf.

[9] Mark Giesbrecht, George Labahn, and Wen-shin Lee. 2004. Symbolic-Numeric Sparse Polynomial Interpolation in Chebyshev Basis and Trigonometric Interpolation. In *Proc. Workshop on Computer Algebra in Scientific Computation (CASC)*. 195–205. https://cs.uwaterloo.ca/~mwg/files/triginterp.pdf.

[10] M. Giesbrecht, G. Labahn, and W. Lee. 2006. Symbolic-numeric Sparse Interpolation of Multivariate Polynomials. In *Proceedings of the 2006 International Symposium on Symbolic and Algebraic Computation (ISSAC '06)*. ACM, New York, NY, USA, 116–123. https://doi.org/10.1145/1145768.1145792

[11] E. Imamoglu and E. L. Kaltofen. 2018. On Computing The Degree Of A Chebyshev Polynomial From Its Value. Manuscript. (May 2018). 10 pages.

[12] E. Kaltofen. 1985. Fast parallel absolute irreducibility testing. *J. Symbolic Comput.* 1, 1 (1985), 57–67. Misprint corrections: *J. Symbolic Comput.* vol. 9, p. 320 (1989). URL: EKbib/85/Ka85_jsc.pdf.

[13] E. Kaltofen. 1985. Sparse Hensel lifting. In *EUROCAL 85 European Conf. Comput. Algebra Proc. Vol. 2 (Lect. Notes Comput. Sci.)*, B. F. Caviness (Ed.). Springer Verlag, Heidelberg, Germany, 4–17. Proofs in [14]. URL: EKbib/85/Ka85_eurocal.pdf.

[14] E. Kaltofen. 1985. *Sparse Hensel lifting*. Technical Report 85-12. Rensselaer Polytechnic Instit., Dept. Comput. Sci., Troy, N. Y. URL: EKbib/85/Ka85_techrep.pdf.

[15] E. Kaltofen. 1994. Asymptotically fast solution of Toeplitz-like singular linear systems. In *Proc. 1994 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'94)*. ACM Press, New York, N. Y., 297–304. Journal version in [16]. URL: EKbib/94/Ka94_issac.pdf.

[16] E. Kaltofen. 1995. Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems. *Math. Comput.* 64, 210 (1995), 777–806. URL: EKbib/95/Ka95_mathcomp.pdf.

[17] Erich L. Kaltofen. 2010. Fifteen years after DSC and WLSS2 What parallel computations I do today [Invited Lecture at PASCO 2010]. In *PASCO'10 Proc. 2010 Internat. Workshop on Parallel Symbolic Comput.*, M. Moreno Maza and Jean-Louis Roch (Eds.). ACM, New York, N. Y., 10–17. URL: EKbib/10/Ka10_pasco.pdf.

[18] 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.)*, P. Gianni (Ed.), Vol. 358. Springer Verlag, Heidelberg, Germany, 467–474. URL: EKbib/88/KaLa88.pdf.

[19] Erich Kaltofen and Wen-shin Lee. 2003. Early Termination in Sparse Interpolation Algorithms. *J. Symbolic Comput.* 36, 3–4 (2003), 365–400. URL: EKbib/03/KL03.pdf.

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

[21] C. P. Pohlig and M. E. Hellman. 1978. An improved algorithm for computing logarithms over GF(*p*) and its cryptographic significance. *IEEE Trans. Inf. Theory* IT-24 (1978), 106–110.

[22] D. Potts and M. Tasche. 2014. Sparse polynomial interpolation in Chebyshev bases. *Linear Algebra and Applic.* 441 (2014), 61–87.

[23] 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.

[24] Ali H. Sayed and Thomas Kailath. 1995. A Look-Ahead Block Schur Algorithm for Toeplitz-Like Matrices. *SIAM J. Matrix Anal. Appl.* 16, 2 (1995), 388–414. https://doi.org/10.1137/S0895479892232649