Error-Correcting Sparse Interpolation in the Chebyshev Basis

Andrew Arnold
Symbolic Computation Group
University of Waterloo
Waterloo, Ontario, Canada
a4arnold@uwaterloo.ca
www.andrewarnold.ca

Erich L. Kaltofen
Department of Mathematics,
North Carolina State University,
Raleigh, North Carolina 27695-8205, USA
kaltofen@math.ncsu.edu
www.math.ncsu.edu/ kaltofen

ABSTRACT

We present an error-correcting interpolation algorithm for a univariate black-box polynomial that has a sparse representation using Chebyshev polynomials as a term basis. Our algorithm assumes that an upper bound on the number of erroneous evaluations is given as input. Our method is a generalization of the algorithm by Lakshman and Saunders [SIAM J. Comput., vol. 24 (1995)] for interpolating sparse Chebyshev polynomials, as well as techniques in error-correcting sparse interpolation in the usual basis of consecutive powers of the variable due to Comer, Kaltofen, and Pernet [Proc. ISSAC 2012, 2014]. We prove the correctness of our list-decoder-based algorithm with a Descartes-rule-of-signs-like property for sparse polynomials in the Chebyshev basis. We show that this list decoder requires fewer evaluations than a naive *majority-rule* block decoder in the case when the interpolant is known to have at most two terms. We also give a new algorithm that reduces sparse interpolation in the Chebyshev basis to that in the power basis, thus making the many techniques for the sparse interpolation in the power basis, for instance, supersparse (lacunary) interpolation over large finite fields, available to interpolation in the Chebyshev basis. Furthermore, we can customize the randomized early termination algorithms from Kaltofen and Lee [J. Symb. Comput., vol. 36 (2003)] to our new approach.

Categories and Subject Descriptors

I.1.2 [**Symbolic and Algebraic Manipulation**]: Algorithms; E.4 [**Coding and Information Theory**]: Error control codes

General Terms: Algorithms.

Keywords: sparse polynomial interpolation; Prony's algorithm; Chebyshev polynomials; Descartes' rule of signs; orthogonal basis; error-correcting code.

ISSAC'15, July 6–9, 2015, Bath, United Kingdom.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3435-8/15/07 ...$15.00.
DOI: http://dx.doi.org/10.1145/2755996.2756652.

1. INTRODUCTION

The sparse univariate interpolation problem is to reconstruct a polynomial $f(x)$ that can be sparsely represented in a given term basis from a number of evaluations that is proportionate to the number of terms t with non-zero coefficients, not the degree of the polynomial. The problem distinguishes the case where the distinct elements at which the polynomial is evaluated are chosen by the algorithm, and the case where the evaluation points cannot be adapted to the interpolation algorithm. The second case constitutes a computationally much harder problem [4], which we will not consider further in this paper. In the first case we think of the polynomial as a black-box function that can be arbitrarily probed.

We consider a black-box polynomial $f(x)$ that can be written as a t-sparse linear combination of Chebyshev polynomials

$$f(x) = \sum_{j=1}^{t} c_j T_{\delta_j}(x) \in \mathsf{K}[x], \quad 0 \le \delta_1 < \delta_2 < \cdots < \delta_t, \quad (1)$$

where K is a field of characteristic $\neq 2$, $c_j \neq 0$ for $1 \le j \le t$, and $T_n \in \mathsf{K}[x]$ is the n-th Chebyshev polynomial of the first kind, defined by:

$$T_0(x) = 1, \quad T_1(x) = x, \quad (2)$$

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \ \text{ for } \ n \ge 2. \quad (3)$$

Since $\deg(T_n) = n$ the set of Chebyshev polynomials forms a (vector-space) basis for $\mathsf{K}[x]$. We will use the following properties of Chebyshev polynomials throughout.

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

i. $\begin{bmatrix} 0 & 1 \\ -1 & 2x \end{bmatrix}^n \begin{bmatrix} 1 \\ x \end{bmatrix} = \begin{bmatrix} T_n(x) \\ T_{n+1}(x) \end{bmatrix}$.

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

iii. $T_m(x)T_n(x) = \frac{1}{2}(T_{m+n}(x) + T_{|m-n|}(x))$.

iv. $T_n(\frac{x+x^{-1}}{2}) = \frac{x^n+x^{-n}}{2}$ *for all $n \ge 0$, as an identity in the function field $\mathsf{K}(x)$.*

v. $T_n(x) = \frac{1}{2}\left(\left(x - \sqrt{x^2-1}\,\right)^n + \left(x + \sqrt{x^2-1}\,\right)^n\right)$, *as an identity in the quadratic extension of the function field $\mathsf{K}(x)$.*

vi. *For $\mathsf{K} = \mathbb{R}$ and $\xi \le -1$ or $\xi \ge 1$, $T_m(\xi) \neq 0$.*

Fact 1.1.i allows the evaluation of T_n at elements from a finite field of characteristic $\neq 2$ in $O(\log(n))$ operations by repeated squaring.

We seek to determine t, the term degrees δ_j, and the coefficients c_j from evaluations $a_i = f(\omega_i)$ where our algorithm chooses which ω_i's to use. Our objective is to require as few ω_i's as possible, and additionally allow for some of the evaluations to be incorrect, that is $a_{\lambda_\kappa} \neq f(\omega_{\lambda_\kappa})$, where λ_κ with $1 \leq \kappa \leq k$ are the indices of the error locations. No algorithm can work without having some bounds: D for the degree, $D \geq \delta_t$; B for the sparsity, $B \geq t$; and $E \geq k$ for the number of errors in the input.

1.1 Organization of paper

In Section 2 we give an overview of sparse interpolation algorithms. We also discuss previous work on error-correcting sparse interpolation in the power (i.e., monomial) basis. Furthermore, we give a simple identity test for sparse polynomials with real coefficients in the Chebyshev basis, relying on a generalization of Descartes' rule of signs [23]. This identity test allows us to verify an interpolant produced by a list-decoding interpolation procedure, such that we can identify the true interpolant.

In Section 3 we generalize the Chebyshev-basis sparse interpolation algorithm from [21]. This is for the purposes of adapting interpolation in this setting to list decoding, following previous work on list-decoding interpolation in the monomial basis [20]. We show that this gives an error-correcting interpolation procedure that requires fewer evaluations than naive "majority rule" decoding in the cases $B = 1, 2$. In Section 4 we present an alternate approach to sparse univariate polynomial interpolation in Chebyshev basis. The Fact 1.1.iv allows a reduction of the problem of interpolating a polynomial that is sparse in Chebyshev basis to interpolating a sparse Laurent polynomial in the power basis, that is, a polynomial with terms x^δ where δ can be a negative integer. We show this algorithm may be adapted to use *early termination*, such that the algorithm can probabilistically determine t and interpolate f from $2t+2$ evaluations in the case when t is not supplied as an input.

Conclusions and discussion of future work are given in Section 5.

2. PRELIMINARIES

The algorithm by Lakshman and Saunders in [21] interpolates $f(x) \in \mathbb{R}[x]$, f given by (1), in the absence of errors. Their algorithm, given a sparsity upper bound B interpolates $f(x)$ from evaluation points $\omega_i = T_i(\xi)$ with $i = 0, 1, \ldots, 2B-1$, for an arbitrary $\xi > 1$. The Monte-Carlo algorithm of Kaltofen and Lee [18] determines t with high probability by randomization, using $\omega_i = T_i(\xi)$ with $i = 0, 1, \ldots, 2t+1$ in the worst case, again without errors. The bounds B and D are needed for guaranteeing an upper bound on the probability of failure. We have, for instance for K a finite field with q elements such that $q > \deg(f)$ and such that $q-1$ has no large prime factor the bit complexity $t^2(\log(q)+\log(D)+\log(B)+\log(t))^{O(1)}$ (see also Section 4 below). In Section 3 we generalize the algorithm of Lakshman and Saunders to interpolate f from sets of evaluation points of the form

$$\omega_{|r+si|}, \quad -B \leq i < 2B,$$

provided the middle B evaluation points $(0 \leq i < B)$ are distinct and $\omega_{|r+si|} > 1$ for all i. This requires τ evaluations, where $2B \leq \tau \leq 3B$, depending on how many indices $|r+si|$ overlap.

2.1 Error-correcting sparse interpolation

Suppose now that for L argument-value pairs (ω_i, a_i) we have $a_i = f(\omega_i)$ for all $i \neq \{\lambda_1, \ldots, \lambda_k\}$ and $k \leq E$ with $0 \leq i \leq L-1$. Here the upper bound E on the number of errors is known on input, but the error locations $\lambda_1, \ldots, \lambda_k$ are not known.

Our algorithms rely on previous work on error-correcting sparse interpolation in the monomial basis. A "majority-rule" interpolation procedure is given in [8]. This is superceded by a procedure based on list-decoding in [20], which is shown to require fewer evaluations to uniquely determine the interpolant.

2.1.1 Majority-rule interpolation

If $L = (2E+1)2B$ we can proceed as in [8]. We interpolate $2E+1$ separate segments of $2B$ argument-value pairs $(T_i(\xi_\ell), a_{i,\ell})$ for $i = 0, 1, \ldots, 2B-1$ and $\ell = 1, 2, \ldots, 2E+1$. If all $T_i(\xi_\ell)$ are distinct and if there are no more than E pairs with $a_{i,\ell} \neq f(T_i(\xi_\ell))$, then the Lakshman-Saunders algorithm [21] applied for each ℓ separately produces the correct sparse interpolant f in Chebyshev basis at least $E+1$ times. By a majority vote we can determine the correct f. As such we refer to this method as *majority-rule* interpolation. The argument distinctness $T_{i_1}(\xi_{\ell_1}) \neq T_{i_2}(\xi_{\ell_2})$ for all $i_1 \neq i_2$ and/or $\ell_1 \neq \ell_2$ can be achieved quickly with high probability by selecting ξ_ℓ uniformly randomly from a sufficiently large finite set. Our model of black-box interpolation with errors presumes that the black box returns a single value, which can be erroneous, and multiple probes to the black box do not reveal erroneous behavior. Surprisingly, in [20] it is shown that $(2E+1)2B$ evaluations is optimal for the Prony/Blahut algorithm: from $(2E+1)2B-1$ pairs one can obtain a second valid sparse interpolant in the power basis.

2.1.2 List-decoding interpolation

Using $E+1$ segments of $2B$ points, one can switch to list decoding (cf. [8, Theorem 3]): at least one of the valid sparse polynomials in Chebyshev basis that is computed by the Lakshman-Saunders algorithm, that is, an interpolant whose number of terms is $\leq B$ and that interpolates $\geq (E+1)2B - E$ argument-value pairs, must agree with the original black box polynomial. For certain inputs one can prove uniqueness (cf. [20, Remark 3]): if $\mathsf{K} = \mathbb{R}$ and $\xi_\ell > 1$ for all ℓ we must have by Corollary 2.4 below a unique valid interpolant.

In [20], the authors give a list-decoding-based sparse univariate polynomial interpolation algorithm for the power basis that requires $L < (E+1)2B$ argument-value pairs (ω_i, a_i), $0 \leq i \leq L-1$, where $\omega_i = \xi^i$ for a suitable $\xi \in \mathsf{K}$. When there are $\leq E$ erroneous values $a_i \neq f(\omega_i)$ the algorithm computes all valid interpolants, which, as stated above, in certain cases are unique. Their idea is to attempt sparse interpolation at the subsequence $(\omega_{r+si}, a_{r+si})$, $i = 0, \ldots, 2B-1$, for all pairs (r, s) with $r \geq 0$ and $s \geq 1$ with $r + (2B-1)s \leq L-1$. If one subsequence avoids all erroneous a_{λ_κ} the sparse interpolant polynomial is produced from the subsequence by the Prony/Blahut algorithm.

In Section 3.2 we transfer the idea to the Chebyshev basis, using the generalization of the Lakshman-Saunders algorithm as a subroutine. In our setting we attempt sparse interpolation at subsequences $(\omega_{|r+si|}, a_{|r+si|})$, for all choices of (r,s), subject to the constraints mentioned in the first paragraph of Section 2.

2.2 A summary of sparse interpolation algorithms

Figure 1 gives a table summarizing sparse univariate polynomial interpolation algorithms in selected bases. The rows show different problem settings and whose columns select different bases. The Pochhammer basis consists of the "falling factorials" $x(x-1)\cdots(x-\delta+1)$, the shifted basis is the variable-shifted power basis $1, x-\sigma, (x-\sigma)^2, \ldots$ with a shift σ that is unknown on input. One may also consider a variable shift in the Chebyshev basis, which is done in [11]. Supersparse algorithms only make sense for coefficients from a finite field and run in time polynomial in log(degree). However, techniques from supersparse interpolation can help stabilize numerical algorithms. Errors seem difficult to correct for sparse polynomials in Pochhammer basis without interpolating the dense polynomial interpolant in power basis by a Reed-Solomon decoder, as does the algorithm for shifted basis in [5]. Interestingly, Blahut's [3] decoder for Reed-Solomon codes uses a sparse interpolation algorithm for error location, which is generalized to multivariate sparse interpolation over $\mathbb{Q}$ in [2]. George Labahn observed the connection between Prony's algorithm in [7] and those algorithms. We do not list our new algorithm from Section 4 in the numerical algorithms row, because we have not conducted the numerical analysis and experiments. Errors can be introduced in the numerical setting, where they are considered outlier evaluations. In [8] such a numerical method is formulated. Algorithms for error correction in the multivariate setting are given in [19].

2.3 Identity testing

Here we present an identity test that will allow us to uniquely identify an interpolant f given by (1), in the case where f is over $\mathsf{K} = \mathbb{R}$. Corollary 4 of [20] uses Descartes' rule of signs to give an identity test for sparse polynomials over $\mathbb{R}$ in the monomial basis (see also [4]). From this it is shown in [20] that one can verify a B-sparse interpolant f from $L = 2B + 2E$ evaluations, provided at most E evaluations are erroneous. Towards a similar result in the Chebyshev basis, we cite a generalization of Descartes' rule of signs, due to Obrechkoff, that gives an upper bound on the number of real roots ≥ 1 for polynomials over $\mathbb{R}$ with a sparse representation in the Chebyshev basis.

Theorem 2.1 (Obrechkoff, 1918) *Define the sequence of polynomials $\{\mathcal{T}_n(x)\}_{n=0}^{\infty}$ by $\mathcal{T}_{-1}(x) = 0$, $\mathcal{T}_0(x) = 1$, and the recurrence relation*

$$x\mathcal{T}_n(x) = \alpha_n \mathcal{T}_{n+1}(x) + \beta_n \mathcal{T}_n(x) + \gamma_n \mathcal{T}_{n-1}(x), \quad n \geq 0,$$

where $\alpha_n, \beta_n, \gamma_n \in \mathbb{R}$, $\alpha_n, \gamma_n > 0$. Let $(c_1, \ldots, c_t)$ be a list of nonzero real numbers with s sign changes between consecutive values, and $0 < \delta_1 < \delta_2 < \cdots < \delta_t \in \mathbb{R}$. Then $\sum_{i=1}^{t} c_i \mathcal{T}_{\delta_i}(x)$ has at most s roots in (ζ_t, ∞), where ζ_t denotes the largest real root of $\mathcal{T}_n$.

See [9] for a proof of Obrechkoff's Theorem. Combined with Fact 1.1.vi this gives the following corollary:

Corollary 2.2 *Let $\mathsf{K} = \mathbb{R}$ and $f(x) = \sum_{i=1}^{t} c_i T_{\delta_i}(x)$, with $\delta_i < \delta_j$ and $c_i \neq 0$ for $1 \leq i < j \leq t$. Then f has at most $t-1$ distinct real roots ≥ 1.*

As $T_n(\xi) > T_m(\xi)$ for $\xi > 1$ and $n > m$, we have in addition the following:

Corollary 2.3 *Let $\xi > 1$ and $f(x)$ be a t-sparse polynomial over $\mathbb{R}$ in the Chebyshev basis. Let $m_1 < m_2 < \cdots < m_B \in \mathbb{Z}_{\geq 0}$ for some $B \geq t$. If $f(T_{m_i}(\xi)) = 0$ for all i, then $f(x)$ is identically zero.*

Corollary 2.4 *Let $\xi > 1$ and $f(x), g(x)$ be two sparse polynomials over $\mathbb{R}$ in the Chebyshev basis, both of sparsity $\leq B$. Let $m_1 < m_2 < \cdots < m_{2B} \in \mathbb{Z}_{\geq 0}$. If $f(T_{m_i}(\xi)) = g(T_{m_i}(\xi))$ for all i, then $f = g$.*

From Corollary 2.3 we have a means of testing whether an interpolant produced by list-decoding interpolation is correct. In particular, if we have L distinct evaluation points $\omega_0, \ldots, \omega_{L-1} > 1$, where $L \geq 2B + 2E$, then a polynomial f comprised of at most B terms in the Chebyshev basis that disagrees with at most E of the evaluations must be the true interpolant.

Corollary 2.3 is also used to show that a linear system given by the algorithm presented in Section 3 gives a unique solution.

3. GENERALIZATION OF THE METHOD OF LAKSHMAN AND SAUNDERS

In this section we develop a generalization of the algorithm given by Lakshman and Saunders in [21] for interpolating a t-sparse polynomial in the Chebyshev basis over the rationals. This generalization will allow us to employ list-decoding interpolation in the Chebyshev basis.

Throughout Section 3 we consider $f \in \mathbb{R}[x]$ of the form given by (1). Let $\xi > 1$ and define the sequence $a_i = f(T_i(\xi))$, for $i \geq 0$. Lemma 3.1 below proves a linear relation of the a_i, taken over indices from the absolute values of an arithmetic progression. This gives us a means of solving for the coefficients. The relation is a straightforward generalization of Lemma 5 in [21], and the proof follows very similarly.

Lemma 3.1 *Fix $s \in \mathbb{Z}_{>0}$ and consider the degree-t polynomial Φ, written in the Chebyshev basis*

$$\Phi(z) = \varphi_t T_t(z) + \varphi_{t-1} T_{t-1}(z) + \cdots + \varphi_0 T_0(z), \quad (4)$$

defined by $\varphi_t = 1$ and $\Phi(T_{s\delta_\ell}(\xi)) = 0$ for $\ell = 1, \ldots, t$. Then, for $i, r \in \mathbb{Z}$,

$$\sum_{j=0}^{t} \varphi_j (a_{|s(i+j)+r|} + a_{|s(i-j)+r|}) = 0. \quad (5)$$

Proof. Observe, using Fact 1.1.ii, that

$$\sum_{j=0}^{t} \varphi_j a_{|s(i+j)+r|} = \sum_{j=0}^{t} \varphi_j \sum_{\ell=1}^{t} c_\ell T_{\delta_\ell}\left(T_{|s(i+j)+r|}(\xi)\right)$$

$$= \sum_{\ell=1}^{t} c_\ell \sum_{j=0}^{t} \varphi_j T_{|s(i+j)+r|}(T_{\delta_\ell}(\xi)). \quad (6)$$

<table>
<tr><th></th><th>monomial</th><th>Chebyshev</th><th>Pochhammer</th><th>shifted power basis</th></tr>
<tr><td rowspan="2">bounded # of terms</td><td rowspan="2">Blahut (1984) [3]</td><td colspan="3">Lakshman, Saunders (1995) [22]</td></tr>
<tr><td rowspan="3">this paper</td><td></td><td>Grigoriev, Karpinski (1993)[16]</td></tr>
<tr><td>supersparse</td><td>Kaltofen (1988) [17]
Garg, Schost (2009) [10]
Arnold, Giesbrecht, Roche (2014) [1]</td><td></td><td>Giesbrecht, Roche (2010) [14]</td></tr>
<tr><td>with errors</td><td>Cormer, Kaltofen, Pernet (2012) [8]
Kaltofen, Pernet (2014) [20]</td><td></td><td>Boyer, Comer, Kaltofen (2014)[8]</td></tr>
<tr><td>early termination</td><td colspan="3">Kaltofen, Lee (2003) [18]</td><td>Giesbrecht, Kaltofen, Lee (2004) [11]</td></tr>
<tr><td>with numerical noise</td><td>Prony (1792)
Giesbrecht, Labahn, Lee (2003) [13]
Giesbrecht, Roche (2011) [15]</td><td>Giesbrecht, Labahn, Lee (2004)[12]</td><td></td><td>Boyer, Comer, Kaltofen (2014) [8]</td></tr>
</table>

Figure 1: Selected sparse univariate interpolation algorithms

Using Facts 1.1.ii and 1.1.iii, we can rewrite the term appearing in the inner sum in (6) as

$$\begin{aligned}&\varphi_j T_{|s(i+j)+r|}\left(T_{\delta_\ell}(\xi)\right)\\ &= \varphi_j\left(2T_{sj}(T_{\delta_\ell}(\xi))T_{|si+r|}(T_{\delta_\ell}(\xi)) - T_{|s(i-j)+r|}(T_{\delta_\ell}(\xi))\right)\\ &= \varphi_j\left(T_jT_{s\delta_\ell}(\xi)) \times 2T_{|si+r|}(T_{\delta_\ell}(\xi)) - T_{|s(i-j)+r|}(T_{\delta_\ell}(\xi))\right).\end{aligned}$$

Thus we can express the inner sum in (6) as

$$\underbrace{\Phi(T_{s\delta_\ell}(\xi))}_{=0} \times 2T_{|si+r|}(T_{\delta_\ell}(\xi)) - \sum_{j=0}^{t}\varphi_j T_{|s(i-j)+r|}(T_{\delta_\ell}(\xi)).$$

Thus (6) becomes

$$\begin{aligned}&-\sum_{\ell=1}^{t} c_\ell\left(\sum_{j=0}^{t}\varphi_j T_{\delta_\ell}(T_{|s(i-j)+r|}(\xi))\right)\\ =&-\sum_{j=0}^{t}\varphi_j\left(\sum_{\ell=1}^{t} c_\ell T_{\delta_\ell}(T_{|s(i-j)+r|}(\xi))\right)\\ =&-\sum_{j=0}^{t}\varphi_j a_{|s(i-j)+r|}.\end{aligned}$$

This gives

$$\sum_{j=0}^{t}\varphi_j a_{|s(i+j)+r|} = -\sum_{j=0}^{t}\varphi_j a_{|s(i-j)+r|}.$$

The identity (5) follows. □

For $r, s \in \mathbb{Z}$, $s \geq 1$ and taking (5) with $i = 0, 1, \ldots, t-1$, this gives us a linear system $A\varphi = -\alpha$, where

$$A = \underbrace{\left[a_{|r+(i+j)s|}\right]_{i,j=0}^{t-1}}_{\text{Hankel matrix}} + \underbrace{\left[a_{|r+(i-j)s|}\right]_{i,j=0}^{t-1}}_{\text{Toeplitz matrix}}, \tag{7}$$

$$\alpha = \left[a_{|r+(i+t)s|} + a_{|r+(i-t)s|}\right]_{i=0}^{t-1}. \tag{8}$$

Thus, provided A is nonsingular, we can obtain φ from A and α. Lemma 3.2 below is an analogue to Lemma 6 of [22]. Our proof is an immediate adaption of theirs.

Lemma 3.2 *Let $r, s \in \mathbb{Z}$, $s > 0$. If the values $|r + si|, 0 \leq i < t$ are distinct, then A is nonsingular.*

PROOF. We will show that $A = UBV$, where

$$U = \left[T_{|r+si|}(T_{\delta_{j+1}}(\xi))\right]_{i,j=0}^{t-1}, \tag{9}$$

$$V = \left[T_{sj}(T_{\delta_{i+1}}(\xi))\right]_{i,j=0}^{t-1}, \tag{10}$$

and B is a diagonal matrix with entries $2c_1, \ldots, 2c_t$. Again using Facts 1.1.ii and 1.1.iii, observe that $(UBV)_{i,j}$ is

$$\begin{aligned}&\sum_{\ell=1}^{t} 2c_\ell T_{|r+is|}(T_{\delta_\ell}(\xi))T_{js}(T_{\delta_\ell}(\xi))\\ &= \sum_{\ell=1}^{t} c_\ell\left(T_{|r+(i+j)s|}(T_{\delta_\ell}(\xi)) + T_{|r+(i-j)s|}(T_{\delta_\ell}(\xi))\right)\\ &= \sum_{\ell=1}^{t} c_\ell\left(T_{\delta_\ell}(T_{|r+(i+j)s|}(\xi)) + T_{\delta_\ell}(T_{|r+(i-j)s|}(\xi))\right)\\ &= a_{|r+(i+j)s|} + a_{|r+(i-j)s|}.\end{aligned}$$

Let b be a row vector such that $bU = \mathbf{0}$. Then

$$\sum_{i=0}^{t-1} b_i T_{|r+si|}(x)$$

is a t-sparse polynomial in the Chebyshev basis with roots $T_{\delta_1}(\xi), \ldots, T_{\delta_t}(\xi)$. By Corollary 2.3, b is necessarily zero. It follows that U is nonsingular. By a similar argument, V is nonsingular. □

3.1 Description of algorithm

We now give a generalization of the sparse interpolation algorithm of Lakshman and Saunders polynomials in the Chebyshev basis given in [21]. Suppose we are given a black-box polynomial $f \in \mathbb{R}[x]$, f of the form (1), with bounds B and D as described in Section 1. For the purposes of Section 3.1 we will assume our evaluations are without errors. First, we choose $\xi > 0$ and $r, s \in \mathbb{Z}$, $s > 0$, such that

$$|r + si| \neq |r + sj| \quad \text{for } 0 \leq i \neq j < B, \tag{11}$$

such that the evaluation points $T_{|r+si|}(\xi)$ are distinct for $0 \leq i < B$.

We query the black-box polynomial f for the evaluations

$$a_{|r+si|} = f(T_{|r+si|}(\xi)), \quad -B \leq i < 2B. \tag{12}$$

This can entail potentially as many as $3B$ evaluations; however, it can be as few as $2B$ in the case that the first or last B evaluations $a_{|r+si|}$ for $-B \leq i < 0$ or $B \leq i < 2B - 1$ are contained in the middle B evaluations $a_{|r+si|}$ for $0 \leq i < B$, e.g., when $r = 0$. The algorithm of Lakshman and Saunders is specifically the case when $r = 0$ and $s = 1$.

If $|r| \geq |r + s(B-1)|$, then we can take $-(r + s(B-1))$ in place of r to obtain the same set of evaluation points (12). Thus without loss of generality we can choose r such that

$|r| \le |r+s(B-1)|$, or equivalently $r \ge -s(B-1)/2$. If $B \ge 3$, the distinctness criterion (11) forces $|r| \ne |r+s(B-1)|$.

Because of the column relation (5) the largest non-singular leading principal submatrix of the Hankel + Toeplitz matrix for all $t' \ge t$

$$\left[a_{|r+(i+j)s|}\right]_{i,j=0}^{t'-1} + \left[a_{|r+(i-j)s|}\right]_{i,j=0}^{t'-1} \qquad (13)$$

is the matrix A in (7) for $t' = t$. By computing the determinant of the leading principal submatrices in increasing order of size, we can thus determine t.

Once we have t we solve the linear system $A\varphi = -\alpha$, where A and α are respectively given by (7) and (8). This gives us the coefficients $\varphi \in \mathbb{R}^{t+1}$ that comprise $\Phi \in \mathbb{R}[z]$ given by (4). We factor Φ to get its t roots, which are exactly the values $T_{\delta_i}(\xi)$ for $i = 1, 2, \ldots, t$. By evaluating $T_j(\xi)$ for appropriate choices of j we can discern δ_i from $T_{\delta_i}(\xi)$ and ξ for each i; see Section 3 in [21].

The coefficients $c_1, \ldots, c_t$ may be given as the solution to the linear system $U^*c = a$, where U^* is the transpose of U given by (9) and $a = (a_{|r|}, a_{|r+s|}, \ldots, a_{|r+(t-1)s|})$. This gives a representation $((c_1, \delta_1), \ldots, (c_t, \delta_t))$ of f.

3.2 A list-decoding interpolation procedure

Now we can consider interpolation in the presence of at most E erroneous evaluations. Our algorithm, on input $B \ge t$ and the sequence of evaluations (12) can determine t and f from a pair (r, s) satisfying (11) with $r \ge -s(B-1)/2$ and $s \ge 1$, provided that the evaluations are correct. We seek such a subsequence of unspoiled evaluations in $a_0, \ldots, a_{L-1}$, where L is computed sufficiently large with respect to the maximum number of errors E, which is also input, to guarantee the existence of that subsequence of at most $3B$ correct evaluations no matter where the error locations $0 \le \lambda_1 < \lambda_2 < \cdots < \lambda_k \le L-1$, with $k \le E$, occur.

We show by example that with sequence locations in arithmetic progression we can decode more errors than with error-free blocks. Since our arithmetic progressions are longer than in [20], $3B$ vs. $2B$, this is not immediately clear. It seems difficult to give an example that can be verified by hand: Let $B = 1$ and $E = 8$: the block method uses $(E+1)2B = 18$ argument-value pairs. However, as is shown in [20, Table 1: $k = 3$, $E = 8$], if in the set of $L = 17$ locations $0, 1, 2, \ldots, 16 = L-1$ we remove any 8 erroneous locations $\lambda_1, \ldots, \lambda_8$, there remains a $k = 3B = 3$-elements arithmetic progression $r, r+s, r+2s \le 16$ for integers $r \ge 0$, $s \ge 1$. Those locations constitute a sequence of the form in (12), even without wrap-around, and our algorithm produces a valid interpolant. For instance, if $\{\lambda_1, \ldots, \lambda_8\} = \{2, 5, 6, 7, 9, 12, 13, 14\}$, which makes blocks of 3 consecutive elements and blocks of 3 consecutive all even or all odd elements impossible, we have the arithmetic progression $\{0, 8, 16\}$ at good locations.

We now show that one such occurrence yields a formula for L for all E sufficiently large. If $L \ge 3B(E+1)$ we have one contiguous segment of $3B$ locations without error using $r = 3\nu B$ and $s = 1$ for some $\nu \ge 0$ (not using wrap-around). We denote by $L_{\min}(3B, E) \le 3B(E+1)$ the minimum length that suffices, without wrap-around ($r \ge Bs$, that is, $r - Bs \ge 0$). As stated in Section 1, from [20, Table 1], we have $L_{\min}(6, 8) \le 34$ ($B = 2, E = 8$).

We now consider at most $E_1 = (E_0+1)m - 1$ errors for some integer $m \ge 1$. We assume that $L_0 \ge L_{\min}(3B, E_0)$. If $L = L_0 m$, then one of the m contiguous segments of L_0 evaluations has $\le E_0$ errors, for otherwise there would be $\ge (E_0+1)m$ errors. That segment has by our assumption for L_0 an arithmetic progression of length $3B$ of locations without errors. Therefore $L_{\min}(3B, (E_0+1)m-1) \le L_0 m$. Since $L_{\min}(3B, E) \le L_{\min}(3B, E_1)$ for $E \le E_1$ we have for $E = (E_0+1)m - (E_0+1) + \nu$ with $\nu = 0, 1, \ldots, E_0$:

$$L_{\min}(3B, (E_0+1)m - (E_0+1) + \nu)$$
$$\le L_{\min}(3B, (E_0+1)m-1) \le L_0 m = L_0 \left\lfloor \frac{E+E_0+1}{E_0+1} \right\rfloor.$$

Our objective is to have $L_0\lfloor (E+E_0+1)/(E_0+1) \rfloor < (E+1)2B$ where the latter is the length required for the block method for E errors.

From the entries in [20, Table 1] we can choose E_0 and L_0 for $B = 1$. If we choose $E_0 = 8$ and $L_0 = 17$, then $17\lfloor (E+9)/9 \rfloor < 2(E+1)$ for all $E \ge 136$. We can also choose $E_0 = 13$ and $L_0 = 23$, such that $23\lfloor (E+14)/14 \rfloor < 2(E+1)$ for all $E \ge 57$.

For $B = 2$: we can choose $E_0 = 8$ and $L_0 = 34$ such that $34\lfloor (E+9)/9 \rfloor < 4(E+1)$ for all $E \ge 136$; or $E_0 = 11$ and $L_0 = 43$ such that $43\lfloor (E+12)/12 \rfloor < 4(E+1)$ for all $E \ge 86$. This gives an algorithm in the case $B = 2$. We produce the evaluations $a_0, \ldots, a_{L-1}$ for $L = 43\lfloor (E+12)/12 \rfloor$, run the algorithm of Section 3.1 over all choices $r, s \in \mathbb{Z}$ with $r \ge -(B-1)s$ and $s \ge 1$, and for each (r, s) check whether the resulting interpolant f agrees with at least $L-E$ evaluations. We are guaranteed a pair (r, s) producing such an f exists, which by Corollary 2.4 must be the true polynomial given by our black-box.

We conjecture that for each $B \ge 3$ there exists an E_0 such that $L_0 = L_{\min}(3B, E_0) < (E_0+1)2B = L_{\text{block}}$. That implies that for all $E > (\rho(E_0+1) - 1)/(1-\rho)$ with $\rho = L_0/L_{\text{block}} < 1$ we have

$$L_0\lfloor (E+E_0+1)/(E_0+1) \rfloor < (E+1)2B. \qquad (14)$$

4. AN ALTERNATE SPARSE CHEBYSHEV INTERPOLATION ALGORITHM

We can reconstruct $f \in \mathsf{K}[x]$ given by (1) when K is a field of characteristic $\ne 2$ and $c_j \ne 0$ for all $1 \le j \le t$, from the evaluations of the form

$$a_i = f\left(\tfrac{\omega^i + \omega^{-i}}{2}\right) \quad \text{for } \omega \in \mathsf{K}, \omega \ne 0, \qquad (15)$$

provided that $\omega^{\delta_j} \ne \omega^{\delta_{j'}}$ for all $0 \le j < j' \le t$. We will show how to interpolate f from a worst-case $2B+1$ evaluations. In Section 4.2 we show how to adapt the algorithm to an early termination scheme, such that only $2t+2$ evaluations $a_0, \ldots, a_{2t+1}$ are required. Note that $a_i = f(T_i(a))$ with $a = (\omega + \omega^{-1})/2$, so the algorithms of the previous sections apply. However, $\omega^{\delta_j} \ne \omega^{\delta_{j'}}$ is not equivalent to $T_{\delta_j}(a) = (\omega^{\delta_j} + \omega^{-\delta_j})/2 \ne (\omega^{\delta_{j'}} + \omega^{-\delta_{j'}})/2 = T_{\delta_{j'}}(a)$. We have from Fact 1.1.iv,

$$g(y) \overset{\text{def}}{=} f\left(\tfrac{y+y^{-1}}{2}\right) = \sum_{j=1}^{t} \frac{c_j}{2}\left(y^{\delta_j} + y^{-\delta_j}\right) \in \mathsf{K}[y, y^{-1}], \qquad (16)$$

which is a $(2t)$-sparse ($(2t-1)$-sparse if $\delta_1 = 0$) Laurent polynomial in power basis. Observe for $g(y)$ in (16) that

$$a_i = g(\omega^i) = a_{-i}, \quad \text{for } i \in \mathbb{Z}. \qquad (17)$$

thus the evaluations $a_0, a_1, \ldots, a_\ell$ give the evaluation of g at ω^i, for $-\ell \le i \le \ell$. By the theory of sparse Prony/Blahut interpolation of Laurent polynomials (see, e.g., [8, Theorem 1]), the sequence of values (15) is linearly generated by the polynomial

$$\Gamma(z) = \prod_{j=1}^{t} \Big((z - \omega^{\delta_j})(z - \omega^{-\delta_j}) \Big). \tag{18}$$

The proof that Γ linearly generates (15) is based on the following Lemma.

Lemma 4.1 *Suppose that the infinite sequence $\{a_i\}_{i \ge 0}$ of elements $a_i \in \mathsf{K}$ is linearly generated by the minimal generator $\Lambda_1(z)$ and the infinite sequence $\{b_i\}_{i \ge 0}$ of elements $b_i \in \mathsf{K}$ is linearly generated by the minimal generator $\Lambda_2(z)$. Then the infinite sequence $\{a_i + b_i\}_{i \ge 0}$ is linearly generated by the least common multiple of Λ_1 and Λ_2, denoted by* $\mathrm{LCM}(\Lambda_1, \Lambda_2)$.

Note that the minimal linear generator $\Lambda(z)$ of (15) can be a non-trivial factor of (18), namely when $\omega^{\delta_j} = \omega^{-\delta_{j'}}$ for $j \ne j'$. Because our assumption $\omega^{\delta_j} \ne \omega^{\delta_{j'}}$ for all $1 \le j < j' \le t$, the factors $z - \omega^{\delta_j}$ occur in the minimal generator for all $1 \le j \le t$. We have the following lemma.

Lemma 4.2 *Let $-\delta_t \le \eta_1, \dots, \eta_\tau \le \delta_t$ be those exponents such that $\omega^{\eta_1}, \dots, \omega^{\eta_\tau}$ form those distinct elements of*

$$\omega^{-\delta_t}, \omega^{-\delta_{t-1}}, \dots, \omega^{-\delta_1}, \omega^{\delta_1}, \dots, \omega^{\delta_t}$$

with

$$c'_\kappa = \Big(\sum_{\substack{\omega^{\delta_j} = \omega^{\eta_\kappa} \\ 1 \le j \le t}} \frac{c_j}{2} \Big) + \Big(\sum_{\substack{\omega^{-\delta_j} = \omega^{\eta_\kappa} \\ 1 \le j \le t}} \frac{c_j}{2} \Big) \ne 0, \quad 1 \le \kappa \le \tau. \tag{19}$$

Then the minimal linear generator of (15) is

$$\Lambda(z) = \prod_{\kappa=1}^{\tau} (z - \omega^{\eta_\kappa}).$$

Proof. If there exists one η_1 satisfying (19), the infinite sequence (15) cannot entirely be a sequence of 0's. The evaluations

$$a_i = \sum_{j=1}^{t} c_j/2\left((\omega^{-\delta_j})^i + (\omega^{\delta_j})^i\right) = \sum_{\kappa=1}^{\tau} c'_\kappa (\omega^{\eta_\kappa})^i \tag{20}$$

are for $0 \le i \le \tau - 1$ the entries of a $\tau \times \tau$ transposed non-singular Vandermonde matrix times a non-zero vector, which cannot be all zero. Therefore, the minimal linear generator Λ of (15) is not the constant polynomial 1. $\Lambda(z)$ is by Lemma 4.1 a factor of $\prod_{\kappa=1}^{\tau}(z - \omega^{\eta_\kappa})$, because $\{c\, \omega^{i\eta_\kappa}\}_{i \ge 1-2t}$ is linearly generated by $z - \omega^{\eta_\kappa}$. Suppose now $\Lambda(z) = \prod_{\kappa=1}^{\tau'}(z - \omega^{\eta_\kappa})$ where $\tau' < \tau$. Since Λ linearly generates all $\{c'_\kappa\, \omega^{i\eta_\kappa}\}_{i \ge 1-2t}$ for $1 \le \kappa \le \tau'$, where $c'_\kappa \ne 0$ is the coefficient corresponding to the exponent η_κ in (20), and by definition linearly generates the sequence (20) for $i \ge 1 - 2t$, Λ must be a linear generator for $\{\sum_{\kappa=\tau'+1}^{\tau} c'_\kappa\, \omega^{i\eta_\kappa}\}_{i \ge 1-2t}$. The latter sequence is linearly generated by $\prod_{\kappa=\tau'+1}^{\tau}(z - \omega^{\eta_\kappa})$ and by a Vandermonde matrix argument the minimal generator $\Lambda^{[2]}(z)$ must be a polynomial factor $\ne 1$. Finally, Λ must be a polynomial multiple of $\Lambda^{[2]}(z)$, which is not possible. Therefore $\tau' = \tau$. □

4.1 Description of algorithm

We compute the minimal generator by a variant of the Berlekamp/Massey algorithm. Suppose on input we have an upper bound on the number of terms, $B \ge t$. One runs the Berlekamp/Massey algorithm on

$$a_{1-2B}, a_{2-2B}, \dots, a_{2B-1}, \alpha,$$

where α is a symbolic value for a_{2B}. If $B > t$ or $\deg(\Lambda) < 2B$, e.g., when $\delta_1 = 0$, a value of α is not needed for computing Λ. The corresponding $(2B) \times (2B)$ Hankel matrix

$$H_B = \left[a_{i+j-2B-1}\right]_{i,j=0}^{2B-1} \tag{21}$$

will then have been identified by the Berlekamp/Massey algorithm as singular. If $2t = \deg(\Lambda) = 2B$, the matrix is identified as non-singular, and Λ is computed as a linear form $\Lambda_\alpha = \Lambda^{[0]} + \alpha\Lambda^{[1]}$. Since Λ then is a reciprocal polynomial, that constraint may determine the value a_{2B} for α, which, for an ω that is selected randomly from a sufficiently large finite set, can be shown to hold with high probability (see Theorem 4.3.ii below). Otherwise, we need to query the black-box polynomial f for the evaluation a_{2B} in order to finish the computation of Λ.

From Λ one computes the exponents $\delta_1, \dots, \delta_t$ as is commonly done in all variants of the Prony/Blahut interpolation algorithm [2, 18, 13, 10, 17], and with the exponents one computes the coefficients c_j from a transposed Vandermonde system.

Remark 4.1 In order to generalize this algorithm for the purposes of error-correction, we would need to consider subsequences of the form α_{r+si}, $i = 1 - 2B, \dots, 2B$, for choices of $r, s \in \mathbb{Z}, s \ge 1$. Similar to the interpolation algorithm in Section 3, this can be as few as $2B + 1$ and as many as $4B$ evaluations, depending on how many evaluations α_{r+si} are used doubly, given $\alpha_j = \alpha_{-j}$ for all j. Moreover, an error at evaluation a_i implies an error at a_{-i}, such that an erroneous evaluation of f can give us an erroneous evaluation of g at possibly two locations.

4.2 Early termination

The above approach has shortcomings compared to the early-termination algorithm in [18]. One needs $2B + 1$ evaluations in the worst case, rather than $2t + 2$. The shortcoming is fixable by adapting the arguments in [18, Proof of Theorem 4]. Let $\alpha_i = g(y^i) = f(\frac{y^i + y^{-i}}{2})$ for $i \in \mathbb{Z}$. We consider the $(2t + 1) \times (2t + 1)$ Hankel matrix with entries in $\mathsf{K}[y, y^{-1}]$,

$$\mathcal{H} = \begin{bmatrix} \alpha_{-2t} & \alpha_{1-2t} & \cdots & \alpha_{-1} & \alpha_0 \\ \alpha_{1-2t} & \alpha_{2-2t} & \iddots & \alpha_0 & \alpha_1 \\ \vdots & \iddots & & \vdots & \vdots \\ \alpha_0 & \cdots & & & \alpha_{2t} \end{bmatrix}. \tag{22}$$

As in [18] we will now show that the square submatrices in the right upper corner are non-singular up to the maximal dimension, with conditions for odd dimensions. By Lemma 4.2 the minimum linear generator $\prod_{j=1}^{t}(z - y^{\delta_j})(z - y^{-\delta_j})$ for $\delta_1 > 0$, and $(z - 1)\prod_{j=2}^{t}(z - y^{\delta_j})(z - y^{-\delta_j})$ for $\delta_1 = 0$, produces a column relation over the field $\mathsf{K}(y)$ so for $\delta_1 > 0$ the upper-right $(2t) \times (2t)$ submatrix of $\mathcal{H}$ is non-singular, and for $\delta_1 = 0$ the right-upper $(2t - 1) \times (2t - 1)$ submatrix of $\mathcal{H}$ is non-singular, and $\mathcal{H}$ is singular.

Theorem 4.3 *Let $\mathcal{H}_i$ be the submatrix of $\mathcal{H}$ formed by the first i rows and the last i columns. Then the following hold:*

i. $\det(\mathcal{H}_i) \neq 0$ *for all even* $i = 2, 4, \ldots, 2t-2$*; if* $\delta_1 > 0$ *then* $\det(\mathcal{H}_{2t}) \neq 0$.

ii. $\det(\mathcal{H}_{2t-1}) \neq 0$*; for all odd* $i = 1, 3, \ldots, 2t-3$*: if* $(\sum_{\nu=1}^{t-(i-1)/2} c_\nu) \neq 0$ *then* $\det(\mathcal{H}_i) \neq 0$.

Proof. Following Lemma 4.2 we denote

$$\begin{array}{lll} \eta_1 = -\delta_t, & \eta_{2t} = \delta_t, & \ell_1 = \ell_{2t} = t, \\ \eta_2 = -\delta_{t-1}, & \eta_{2t-1} = \delta_{t-1}, & \ell_2 = \ell_{2t-1} = t-1, \\ \vdots & \vdots & \vdots \\ \eta_t = -\delta_1, & \eta_{t+1} = \delta_1, & \ell_t = \ell_{t+1} = 1, \end{array}$$

such that $\eta_i = -\delta_{\ell_i}$ for $1 \leq i \leq t$, and $\eta_i = \delta_{\ell_i}$ for $t+1 \leq i \leq 2t$. Let $\beta_\kappa = y^{\eta_\kappa}$ for $1 \leq \kappa \leq 2t$. The matrix $\mathcal{H}_i$ can be factored as (see [18, Eq. (7)])

$$\mathcal{H}_i = \mathcal{B}_i C_{2t} \bar{\mathcal{B}}_i^*, \quad C_{2t} = \operatorname{diag}(c_{\ell_1}/2, c_{\ell_2}/2, \ldots, c_{\ell_{2t}}/2) \quad (23)$$

where the * is the transposition operator, with $\mathcal{B}_i, \bar{\mathcal{B}}_i \in K(y)^{i \times (2t)}$ given by

$$\mathcal{B}_i = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \beta_1 & \beta_2 & \cdots & \beta_{2t} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_1^{i-1} & \beta_2^{i-1} & \cdots & \beta_{2t}^{i-1} \end{bmatrix}, \quad \bar{\mathcal{B}}_i = \begin{bmatrix} \beta_1^{1-i} & \beta_2^{1-i} & \cdots & \beta_{2t}^{1-i} \\ \beta_1^{2-i} & \beta_2^{2-i} & \cdots & \beta_{2t}^{2-i} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_1^{-1} & \beta_2^{-1} & \cdots & \beta_{2t}^{-1} \\ 1 & 1 & \cdots & 1 \end{bmatrix}.$$

Let $M_{J,K}$ be the determinant of the submatrix of M consisting of rows in $J = \{j_1, \ldots, j_i\}$ and columns in $K = \{k_1, \ldots, k_i\}$. As in [18, Proof of Theorem 4], by the Binet-Cauchy formula with $I = \{1, 2, \ldots, i\}$, we can write $\det(\mathcal{H}_i)$ as

$$\begin{aligned} & \sum_J \sum_K (\mathcal{B}_i)_{I,J} (C_{2t})_{J,K} (\bar{\mathcal{B}}_i^*)_{K,I} \\ &= \sum_J 2^{-i} \left(\textstyle\prod_{m=1}^i c_{\ell_{j_m}} \right) (\mathcal{B}_i)_{I,J} (\bar{\mathcal{B}}_i^*)_{J,I} \\ &= \sum_J 2^{-i} \left(\textstyle\prod_{m=1}^i c_{\ell_{j_m}} \beta_{j_m}^{1-i} \right) \det \left(\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \beta_{j_1} & \beta_{j_2} & \cdots & \beta_{j_i} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{j_1}^{i-1} & \beta_{j_2}^{i-1} & \cdots & \beta_{j_i}^{1-i} \end{bmatrix} \right)^2 \\ &= \sum_J 2^{-i} \left(\textstyle\prod_{m=1}^i c_{\ell_{j_m}} \beta_{j_m}^{1-i} \right) \prod_{1 \leq v < u \leq i} (\beta_{j_u} - \beta_{j_v})^2. \qquad (24) \end{aligned}$$

In the expansion of the products in (24) we have the terms $\beta_{j_1}^{2i-2} \beta_{j_2}^{2i-4} \cdots \beta_{j_{i-1}}^2$ so we have a summand term $\beta_{j_1}^{i-1} \beta_{j_2}^{i-3} \times \cdots \beta_{j_{i/2}}^1 \beta_{j_{i/2+1}}^{-1} \cdots \beta_{j_{i-1}}^{3-i} \beta_{j_i}^{1-i}$. For Part i, if we set

$$\begin{array}{ll} j_1 = 2t, & j_i = 1, \\ j_2 = 2t-1, & j_{i-1} = 2, \\ \vdots & \vdots \\ j_{i/2} = 2t - i/2 + 1, & j_{i/2+1} = i/2, \end{array}$$

we obtain in the expansion of (24) the term y^D with $D = 2(i-1)\delta_t + 2(i-3)\delta_{t-1} + \cdots + 2\delta_{t-(i/2-1)}$ for even i. For $i \leq 2t-2$, that is $t - (i/2 - 1) \geq 2$ and $\delta_{t-(i/2-1)} > 0$ and for $i = 2t$ and $\delta_1 > 0$ that y^D term can only occur once in the expansion, and therefore cannot cancel. We add a brief explanation to the last claim. We multiply the sum (24) by $(\beta_1 \cdots \beta_t)^{i-1}$ and expand: the terms are then of the form $\beta_{j_1}^{w_1} \cdots \beta_{j_i}^{w_i} \beta_{j_{i+1}}^{i-1} \cdots \beta_{j_t}^{i-1}$ with $\{j_1, \ldots, j_t\} = \{1, \ldots, t\}$, $w_\nu \geq 0$ and $w_1 + \cdots + w_i = i(i-1)$ and additional constraints. For instance, if $w_1 = 2i-2$ then $w_\nu \leq 2i-4$ for all $\nu \geq 2$. The single maximum of the degree in y is at the assignment $w_1 = 2i-2$ and $\beta_{j_1} = y^{\delta_t}$, $w_i = 0$ and $\beta_{j_i} = y^{-\delta_t}, \ldots,$ $w_{i/2} = i$ and $\beta_{j_{i/2}} = y^{\delta_{t-(i/2-1)}}$, $w_{i/2+1} = i-2$ and $\beta_{j_{i/2}} = y^{-\delta_{t-(i/2-1)}}$, which gives the largest δ_κ the largest available weights, and the smallest $-\delta_\kappa$ the smallest available weights.

For odd i in Part ii, the highest degree term $y^{D'}$ is not unique. We have the summand term in (24)

$$\beta_{j_1}^{i-1} \beta_{j_2}^{i-3} \cdots \beta_{j_{(i-1)/2}}^2 \beta_{j_{(i+1)/2}}^0 \beta_{j_{(i+3)/2}}^{-2} \beta_{j_{(i+5)/2}}^{-4} \cdots \beta_{j_i}^{1-i}.$$

If we set $j_1 = 2t, j_i = 1, j_2 = 2t-1, j_{i-1} = 2, \ldots, j_{(i-1)/2} = 2t - (i-3)/2, j_{(i+3)/2} = (i-1)/2$, and

$$j_{(i+1)/2} \in \{(i+1)/2, (i+3)/2, \ldots, 2t - (i-1)/2\},$$

we obtain in the expansion of (24) the term $y^{D'}$ with $D' = 2(i-1)\delta_t + 2(i-3)\delta_{t-1} + \cdots + 8\delta_{t-((i+1)/2-3)} + 4\delta_{t-((i+1)/2-2)}$, and only with such settings. The coefficient of $y^{D'}$ is $2^{-i} \times c_{\ell_1} c_{\ell_{2t}} \cdots c_{\ell_{(i-1)/2}} c_{\ell_{2t-(i-3)/2}} \sum_{\kappa=(i+1)/2}^{2t-(i-1)/2} c_{\ell_\kappa}$, which, by our assumption of Part ii, is non-zero ($\ell_\kappa = \ell_{2t-\kappa+1} = t - \kappa + 1$ for $1 \leq \kappa \leq t$); for $i = 2t-1$ we have $\sum_{\kappa=(i+1)/2}^{2t-(i-1)/2} c_{\ell_\kappa} = \sum_{\kappa=t}^{t+1} c_{\ell_\kappa} = 2c_1 \neq 0$. (Cf. [18, Proof of Theorem 11].) □

The early termination algorithm selects two random field elements $\omega, \omega' \in S$ uniformly from a sufficiently large finite set of field elements $S \subseteq \mathsf{K}$ and computes the coefficients of Λ by a linear system solver for the Toeplitz matrix

$$Z^{[\infty]} = \begin{bmatrix} a_0 + \omega' & a_1 + \omega' & \cdots & a_{2t-1} + \omega' & a_{2t} + \omega' & \cdots \\ a_{-1} + \omega' & a_0 + \omega' & \ddots & a_{2t-2} + \omega' & a_{2t-1} + \omega' & \cdots \\ \vdots & \ddots & & \vdots & \vdots & \\ a_{1-2t} + \omega' & & \cdots & a_0 + \omega' & a_1 + \omega' & \cdots \\ \vdots & & & \vdots & \vdots & \ddots \end{bmatrix}. \qquad (25)$$

Adding ω' changes f to $f + \omega'$ (cf. [18, Section 3.4]) and constitutes a rank 1 update in H_B in (21) that in Theorem 4.3.ii is shown to make with high probability all 1×1, $2 \times 2, \ldots, \tau \times \tau$ leading principal submatrices non-singular, where $\tau = 2t-1$ if $\delta_1 = 0$, that is if g has a non-zero constant term, and where $\tau = 2t+1$ otherwise. Although t and τ are unknown, one can now use Trench's $O(t^2)$ Toeplitz solver, or the asymptotically faster algorithms in [6] of arithmetic complexity $O(t(\log t)^2 \operatorname{loglog}(t))$, for locating the first singular leading principal submatrix and for computing Λ. The additive constant ω' can be avoided, that is setting $\omega' = 0$ in (25), if one uses Gaussian elimination or a block Toeplitz solver with 2×2 blocks.

Remark 4.2 The argument used for odd i in the proof of Lemma 4.3 relates to an old open question in [18, Footnote in the Proof of Theorem 4], where the authors interpolate the polynomial $f(x_1, \ldots, x_n) = \sum_{j=1}^t c_j x_1^{e_{j,1}} \cdots x_n^{e_{j,n}}$ with $c_j \neq 0$ from $a_i = f(\omega_1^i, \ldots, \omega_n^i)$. If we use the sequence $a_0, a_1, \ldots, a_{2t-1}$, we need to prove for indeterminate variables $y_1, \ldots, y_n$ and term values $\beta_j = y_1^{e_{j,1}} \cdots y_n^{e_{j,n}}$ the non-vanishing of the determinantal polynomial expression, for

$i = 2, 3, \ldots, t$:

$$0 \neq \sum_{J=\{j_1,\ldots,j_i\}} c_{j_1} \cdots c_{j_i} \cdot \det\left(\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \beta_{j_1} & \beta_{j_2} & \cdots & \beta_{j_i} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{j_1}^{i-1} & \beta_{j_2}^{i-1} & \ldots & \beta_{j_i}^{i-1} \end{bmatrix}\right)^2$$
$$= \sum_{J=\{j_1,\ldots,j_i\}} c_{j_1} \cdots c_{j_i} \cdot \prod_{1\le v<u\le i} (\beta_{j_u} - \beta_{j_v})^2. \quad (26)$$

We assume that the β_j are ordered lexicographically $\beta_1 \succ \ldots \succ \beta_t$. The difficulty is that the highest ordered term in the expansion of the products in (26) is $\beta_1^{2i-2} \cdots \beta_{i-1}^2$ and occurs with coefficients $c_1 \cdots c_{i-1} c_\nu$ where $\nu = i, \ldots, t$. As in the proof of Lemma 4.3, we may make $(\sum_{\nu=i}^{t} c_\nu) \neq 0$ by adding a random constant to f, that is, to a_i, but then we may have $t+1$ terms. In [18, Section 3.2] the sequence is shifted by one index to $a_1, \ldots, a_{2t}$ in order to avoid the increase in the number of terms. Because we exploit the symmetry of the evaluation points in (21), we cannot utilize that shift for interpolating the Laurent polynomial (16) (see also [18, Proof of Theorem 11]). However, our Theorem 4.3.i yields early termination on all submatrices of even dimensions in any case. □

5. CONCLUSIONS AND FUTURE WORK

We presented two methods for the black-box interpolation of a sparse polynomial f in the Chebyshev basis. The first approach was a generalization of the Lakshman-Saunders algorithm [21]. This was used as part of a list-decoding interpolation procedure, which was shown to be better than majority-rule interpolation when the interpolant is known to have at most two terms. In the case that $f \in \mathbb{R}[x]$, it is shown that list-decoding interpolation will identify f.

The other interpolation procedure reduces the problem of sparse interpolation in the Chebyshev basis to sparse interpolation of a Laurent polynomial in the monomial basis. This was adapted to early termination, such that we can probabilistically determine t and f from $2t+2$ evaluations.

We conjecture that, for any sparsity bound B and sufficiently large error bound E, one can perform error-correcting interpolation with fewer than the bound of $(2B+1)E$ errors, a naive bound due to majority-rule interpolation. We also hope to adapt our new interpolation algorithm of Section 4 to error correction.

6. ACKNOWLEDGEMENTS

We would like to thank the anonymous referees for their constructive comments, and for bringing [9] to our attention. This research was supported in part by NSERC (Arnold) and the National Science Foundation under Grants CCF-1115772 and CCF-1421128 (Kaltofen).

Note added July 26, 2015: Included the condition (19) in Lemma 4.2 and slightly adjusted the proof.

Clément Pernet has provide the entry $n_{9,12} = 74$ for [20, Table 1], which (see end of Section 3.2) for $B = 3$ provides an interpolant from $74\lceil (E+13)/13 \rceil < 6(E+1)$ evaluations with $\le E$ errors for all $E \ge 222$.

Note added May 31, 2016: Corrected (24).

7. REFERENCES

[1] A. Arnold, M. Giesbrecht, and D. S. Roche. Sparse interpolation over finite fields via low-order roots of unity. In *Proc. 39th Internat. Symp. Symbolic Algebraic Comput.*, ISSAC '14, pages 27–34. ACM, 2014.

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

[3] R. E. Blahut. A universal Reed-Solomon decoder. *IBM J. Res. Develop.*, 18(2):943–959, Mar. 1984.

[4] A. Borodin and P. Tiwari. On the decidability of sparse univariate polynomial interpolation. *Computational Complexity*, 1:67–90, 1991.

[5] B. Boyer, M. Comer, and E. Kaltofen. Sparse polynomial interpolation by variable shift in the presence of noise and outliers in the evaluations. In R. Feng, W.-s. Lee, and Y. Sato, editors, *Computer Mathematics*, pages 183–197. Springer Berlin Heidelberg, 2014.

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

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

[8] M. T. Comer, E. L. Kaltofen, and C. Pernet. Sparse polynomial interpolation and berlekamp/massey algorithms that correct outlier errors in input values. In *Proc. 37th Internat. Symp. Symbolic Algebraic Comput.*, ISSAC '12, pages 138–145. ACM, 2012.

[9] D. K. Dimitrov and F. R. Rafaeli. Descartes' rule of signs for orthogonal polynomials. *East J. Approx.*, 15(2):233–262, 2009.

[10] S. Garg and Éric. Schost. Interpolation of polynomials given by straight-line programs. *Theoretical Comput. Sci.*, 410(27-29):2659–2662, 2009.

[11] M. Giesbrecht, E. Kaltofen, and W.-s. Lee. Algorithms for computing sparsest shifts of polynomials in power, chebyshev, and pochhammer bases. *J. Symb. Comput.*, 36(3-4):401–424, 2003.

[12] M. Giesbrecht, G. Labahn, and W. Lee. Symbolic-numeric sparse polynomial interpolation in Chebyshev basis and trigonometric interpolation. In *Proc. Workshop on Computer Algebra in Scientific Computation (CASC)*, pages 195–205, 2004. https://cs.uwaterloo.ca/~mwg/files/triginterp.pdf.

[13] M. Giesbrecht, G. Labahn, and W. Lee. Symbolic-numeric sparse interpolation of multivariate polynomials. *J. Symbolic Comput.*, 44:943–959, 2009.

[14] M. Giesbrecht and D. Roche. Interpolation of shifted-lacunary polynomials. *Computational Complexity*, 19(3):333–354, 2010.

[15] M. Giesbrecht and D. Roche. Diversification improves interpolation. In A. Leykin, editor, *Proc. 36th Internat. Symp. Symbolic Algebraic Comput.*, pages 123–130, New York, N. Y., 2011. Association for Computing Machinery.

[16] D. Y. Grigoriev and M. Karpinski. A zero-test and an interpolation algorithm for the shifted sparse polynomials. In *Proc. AAECC-10*, volume 673 of *Lect. Notes Comput. Sci.*, pages 162–169. Springer Verlag, 1993.

[17] E. Kaltofen. Fifteen years after DSC and WLSS2 What parallel computations I do today [Invited lecture at PASCO 2010]. In *Proc. 2010 Internat. Workshop on Parallel Symbolic Comput.*, PASCO '10', pages 10–17, 2010. URL: http://www.math.ncsu.edu/~kaltofen/bibliography/10/Ka10_pasco.pdf.

[18] E. Kaltofen and W.-s. Lee. Early termination in sparse interpolation algorithms. *Journal of Symbolic Computation*, 36(3):365–400, 2003.

[19] E. Kaltofen and Z. Yang. Sparse multivariate function recovery with a high error rate in evaluations. In K. Nabeshima, editor, *Proc. 39th Internat. Symp. Symbolic Algebraic Comput.*, pages 280–287, New York, N. Y., 2014. Association for Computing Machinery. URL: http://www.math.ncsu.edu/~kaltofen/bibliography/14/KaYa14.pdf.

[20] E. L. Kaltofen and C. Pernet. Sparse polynomial interpolation codes and their decoding beyond half the minimum distance. In *Proc. 39th Internat. Symp. Symbolic Algebraic Comput.*, ISSAC '14, pages 272–279. ACM, 2014.

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

[22] Lakshman Y. N. and B. D. Saunders. Sparse shifts for univariate polynomials. *Applic. Algebra Engin. Commun. Comput.*, 7(5):351–364, 1996.

[23] N. Obrechkoff. On the roots of algebraic equations. *Annuaire University of Sofia Phys.-Math. Fac.*, 19:43–76, 1923.