Journal of Symbolic Computation 36 (2003) 401–424

Journal of
Symbolic
Computation

www.elsevier.com/locate/jsc

Algorithms for computing sparsest shifts of polynomials in power, Chebyshev, and Pochhammer bases

Mark Giesbrecht[a,*], Erich Kaltofen[b], Wen-shin Lee[a]

[a] *School of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*
[b] *Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, USA*

Received 23 November 2002; accepted 20 March 2003

Abstract

We give a new class of algorithms for computing sparsest shifts of a given polynomial. Our algorithms are based on the early termination version of sparse interpolation algorithms: for a symbolic set of interpolation points, a sparsest shift must be a root of the first possible zero discrepancy that can be used as the early termination test. Through reformulating as multivariate shifts in a designated set, our algorithms can compute the sparsest shifts that simultaneously minimize the terms of a given set of polynomials. Our algorithms can also be applied to the Pochhammer and Chebyshev bases for the polynomials, and potentially to other bases as well. For a given univariate polynomial, we give a lower bound for the optimal sparsity. The efficiency of our algorithms can be further improved by imposing such a bound and pruning the highest degree terms.

Keywords: Sparse shifts; Early termination; Sparse polynomial; Sparse interpolation; Chebyshev basis; Pochhammer basis

1. Introduction

Let $f(x_1, \ldots, x_n) = \sum_{i=1}^{r} u_i x_1^{d_{i,1}} \cdots x_n^{d_{i,n}} \in \mathsf{D}[x_1, \ldots, x_n]$ be a multivariate polynomial whose coefficients are in an integral domain D. A sparsest shift within S is a vector $(\theta_1, \ldots, \theta_n) \in S$ such that the number of (shifted) terms is minimized as τ in

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{\tau} \gamma_i (x_1 + \theta_1)^{\delta_{i,1}} \cdots (x_n + \theta_n)^{\delta_{i,n}} \qquad \text{and} \qquad \gamma_i \neq 0.$$

[*] Corresponding author. Tel.: +1-519-888-4567; fax: +1-519-888-1208.
E-mail addresses: mwg@uwaterloo.ca (M. Giesbrecht), kaltofen@math.ncsu.edu (E. Kaltofen), ws2lee@scg.uwaterloo.ca (W.-s. Lee).
URLs: http://www.uwaterloo.ca/~mwg, http://www.kaltofen.us, http://www.wen-shin.com.

0014-5793/03/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0747-7171(03)00087-7

The sparsest shifts might not be unique: for $f = x^2 + x + 1 \in \mathsf{D}[x]$ and $\overline{\mathsf{K}}$ the algebraic closure of D, there are three sparsest shifts in $\overline{\mathsf{K}}$ with $\tau = 2$, namely $\theta = -1/2$, $\theta = \rho_1$ and ρ_2 where $f = (x - \rho_1)(x - \rho_2)$. However, in the univariate case Lakshman and Saunders (1996) give a sufficient condition for the uniqueness of the sparsest shift. While Grigoriev and Lakshman (2000) give some generalizations of the uniqueness properties in the multivariate case, we provide a stronger result in the univariate case.

Sparse shifts can dramatically reduce the size of a symbolic expression. A classical example, by Joel Moses, is $\int 1+(x+1)^n \mathrm{d}x = x+(x+1)^{n+1}/(n+1)$. Sparse shifts can be useful when interpolating the black-box polynomial outputs of the algorithms in Kaltofen and Trager (1990), say the black box for the irreducible factors of a matrix determinant with symbolic entries. It is possible that a sparse shift can make a factor manageable, while the standard representation, in Knuth's (1997) words, "would fill the universe". Algorithms for computing a sparse shift could therefore be considered simplification tools.

We give a new class of algorithms for efficiently computing the sparsest shifts. Our algorithms are based on the early termination version of sparse interpolation algorithms (Kaltofen et al., 2000; Kaltofen and Lee, 2003), which capture the sparsity of the target polynomial in a designated basis when the early termination occurs. The main idea is that for a *symbolic* set of interpolation points, a shift must be a root of a discrepancy that is used as the early termination test; a sparsest shift is the first such zero to occur. We note that our approach is similar to that of Grigoriev and Karpinski (1993), who use Wronskians (Grigoriev et al., 1994) in place of discrepancies. Here we can assume that the input polynomial f is being interpolated and we are given a black-box procedure for its evaluation. For coefficient fields of small cardinality we require that the black box allows evaluations on points from an extension field (Grigoriev et al., 1990), which can be realized in a computer program as the so-called extended domain black-box object (Díaz and Kaltofen, 1998). We note that for efficiency it is sometimes useful to compute the coefficients of f via interpolation before employing our methods.

Through randomization we can dramatically improve the efficiency of our algorithms. Our randomization is of the Las Vegas kind—always correct and probably fast—because one may always check a candidate sparsest shift via a sparse interpolation algorithm. First, we may choose random values as interpolation points rather than symbolic ones, and employ the probabilistic analysis of DeMillo and Lipton (1978), Zippel (1979) and Schwartz (1980). In the univariate case, and in the multivariate case where a very sparse shift exists, we may replace the polynomial root finder by a GCD procedure. This is possible since the sparsest shifts are the roots of a sequence of discrepancies. For the sparsest shifts in the power bases, we can provide a complete probabilistic analysis when the algorithm is run on two independent trials or when all discrepancies up to $2\deg f$ are considered. For univariate shifts within $\mathbb{Q}$, we can further eliminate the indeterminate shift variable in our algorithm by evaluating at random integers such that the shift is determined through a large prime factor. We can provide proof for a method that uses ten independent trials with the provision that the sparsest shift is unique.

The running times of our methods compare favorably with the previously best algorithms (Grigoriev and Karpinski, 1993; Grigoriev and Lakshman, 1995; Lakshman and Saunders, 1996; Grigoriev and Lakshman, 2000). Not accounting for the length of the intermediately computed scalars, our method at its best, in the univariate rational case when no symbolic value for the shift is carried along, requires $O(\tau^2)$ operations and $O(\tau)$ evaluations of f. When the bit-lengths of the rational numbers involved are considered, our algorithm requires $O(\tau^2\mathsf{M}(\tau^2 \deg(f) \log\|f\|))$ bit operations, where $O(\mathsf{M}(\ell))$ bit operations are sufficient to multiply two integers with ℓ bits (and $\mathsf{M}(\ell) = \ell^2$ using the standard algorithm, and $\mathsf{M}(\ell) = \ell \log \ell \log\log \ell$ using asymptotically fast arithmetic). The algorithm of Lakshman and Saunders (1996) uses $O(\tau^2 \deg f + \tau^5)$ arithmetic operations and $4\tau + 2$ values of f and its derivatives. We note that Grigoriev and Karpinski (1993) have established the problem to be in polynomial-time. Over a general field (supporting root finding), our "one projection" algorithm of Section 3.2 requires $O(\tau^2\mathsf{M}(\tau \deg f))$ operations in K. $O(\tau)$ evaluations of f at symbolic points, or $O(\tau \deg f)$ evaluations at points in K are also required.

We can also find the sparsest shifts of a set of polynomials by reformulating the problem as finding the multivariate sparsest shift within a designated set. The efficiency of our algorithms can be further improved by constraining the computations within the bounds, whenever available, for the optimal sparsity, and by pruning the highest degree terms which remain unchanged in all shifts.

2. Sparse interpolations and sparsity in shifted bases

2.1. Sparse interpolations in any given power basis

Given a black-box polynomial $f(x_1, \ldots, x_n) \in \mathsf{D}[x_1, \ldots, x_n]$ in the power basis generated by $x_1, \ldots, x_n$.

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{r} u_i x_1^{d_{i,1}} \cdots x_n^{d_{i,n}}, \qquad u_i \in \mathsf{D}\backslash\{0\}, \tag{1}$$

under another power basis of $w_j = a_{j,0} + a_{j,1}x_1 + \cdots + a_{j,n}x_n (1 \le j \le n)$ with $a_{j,i} \in \mathsf{D}$ for $1 \le j \le n$, f is represented as:

$$g(w_1, \ldots, w_n) = \sum_{i=1}^{t} c_i w_1^{e_{i,1}} \cdots w_n^{e_{i,n}}, \qquad c_i \in \mathsf{K}\backslash\{0\}, \tag{2}$$

where K is the quotient field of D. Here, t, $e_{i,j}$, and c_i are all dependent on the definition of w_j; the enumeration in i depends on the term order being used. The representation in (1) is a special case of (2).

The sparsity of a polynomial depends on the choice of basis in the representation; we consider the sparse interpolations in the power basis of w_j for $1 \le j \le n$.

The black box f takes values for each x_j as input. In order to interpolate f in w_j, namely $g(w_1, \ldots, w_n)$ in (2), we need to form a black box for g that takes inputs as values for w_j such that $f(x_1, \ldots, x_n) = g(w_1, \ldots, w_n)$. By definition,

$$\underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}}_{W} = \underbrace{\begin{bmatrix} a_{1,0} \\ a_{2,0} \\ \vdots \\ a_{n,0} \end{bmatrix}}_{A_0} + \underbrace{\begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{X} \tag{3}$$

and we have $X = A^{-1}(W - A_0)$. We may assume the matrix A is non-singular because both x_j and w_j are bases of $\mathsf{K}[x_1, \dots, x_n]$. Let $g(W^{\mathrm{Tr}})$ denote $g(w_1, \dots, w_n)$. A black box for g can be constructed by evaluating f at $(A^{-1}(W - A_0))^{\mathrm{Tr}}$:

$$g(W^{\mathrm{Tr}}) = f((A^{-1}(W - A_0))^{\mathrm{Tr}}), \tag{4}$$

with both A^{-1} and A_0 obtained from the given w_j. By applying the Ben-Or/Tiwari algorithm (Ben-Or and Tiwari, 1988) and its early termination (Kaltofen et al., 2000; Kaltofen and Lee, 2003) to the black box $g(w_1, \dots, w_n)$ in (4), we establish the corresponding sparse interpolations of f in w_j.

2.2. *The sparsity of polynomials in shifted bases*

Consider a univariate polynomial $f(x) \in \mathsf{D}[x]$ in two different power bases: x and $(x + s)$ with $s \neq 0$. Let $d = \deg f(x)$, $u_i \neq 0$ for $1 \leq i \leq r$, and $c_j \neq 0$ for $1 \leq j \leq t$, then f can be represented as:

$$f(x) = u_1 x^{d_1} + u_2 x^{d_2} + \cdots + u_r x^{d_r} \tag{5}$$

$$= c_1(x + s)^{e_1} + c_2(x + s)^{e_2} + \cdots + c_t(x + s)^{e_t}, \tag{6}$$

with $d_1 < d_2 < \cdots < d_r = d$ and $e_1 < e_2 < \cdots < e_t = d$. The number of terms of f in the basis of x is r, and in the basis of $(x + s)$ is t.

As a special case of multivariate sparsifying transformations, Grigoriev and Lakshman (2000) gave an inequality between the sparsities in different shifted bases. Using a different method, we give a stronger result in the univariate case.

Theorem 1. *For a univariate polynomial f with $\deg f = d$, represented in any two different bases with number of term r and t respectively, $r + t > d + 1$, provided $\binom{d}{j} \neq 0$ for all $0 < j < d$ when computed as an element in D.*

Proof. Since the indeterminate x in (5) can be used to represent a shifted basis itself, by using the representations in (5) and (6) in our proof, we will not lose generality.

If $r = d + 1$ in (5), since $t \geq 1$ in (6), we have $r + t > d + 1$.

When $r < d + 1$, there are $\kappa = d + 1 - r$ many terms of f in (5) with a coefficient of zero. Let their degrees be ordered as $\delta_1 > \delta_2 > \cdots > \delta_\kappa$. We expand (6) and collect the coefficient for each x^{δ_i}, which are contributed from all terms of degree no less than δ_i. In other words, collect all $c_j(x + s)^{e_j}$ with $e_j \geq \delta_i$ and

$$\binom{e_t}{\delta_i} s^{e_t - \delta_i} c_t + \binom{e_{t-1}}{\delta_i} s^{e_{t-1} - \delta_i} c_{t-1} + \cdots + \binom{e_j}{\delta_i} s^{e_j - \delta_i} c_j = 0.$$

As a result, we have the following system for $1 \le i \le \kappa$:

$$\underbrace{\begin{bmatrix} \binom{e_t}{\delta_1} s^{e_t-\delta_1} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & & & \vdots \\ \binom{e_t}{\delta_i} s^{e_t-\delta_i} & \cdots & \binom{e_j}{\delta_i} s^{e_j-\delta_i} & \cdots & 0 \\ \binom{e_t}{\delta_{i+1}} s^{e_t-\delta_{i+1}} & \cdots & \binom{e_j}{\delta_{i+1}} s^{e_j-\delta_{i+1}} & \cdots & \\ \vdots & \ddots & \vdots & \ddots & \\ \binom{e_t}{\delta_\kappa} s^{e_t-\delta_\kappa} & \cdots & \binom{e_j}{\delta_\kappa} s^{e_j-\delta_\kappa} & \cdots & \binom{e_1}{\delta_\kappa} s^{e_1-\delta_\kappa} \end{bmatrix}}_{\mathcal{V}} \begin{bmatrix} c_t \\ \vdots \\ c_{j+1} \\ c_j \\ \vdots \\ c_1 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

In the matrix $\mathcal{V}$, we consider the ith row V_i and its number of non-zero entries v_i. Note that the first v_i entries in V_i are non-zero and $v_i \le v_{i+1}$. We want to claim V_i is not a linear combination of $V_1, \ldots, V_{i-1}$ and $v_i \ge i+1$ for $1 < i < t$.

We have $v_1 \ge 2$, otherwise $\binom{e_t}{\delta_1} s^{e_t-\delta_1} c_t = 0$ implies $c_t = 0$. If $v_1 = v_2 = 2$, the non-zero part of V_1 and V_2 form a 2×2 transpose Vandermonde-like system (cf. Evans and Isaacs, 1976) of rank 2 with solution $c_t = c_{t-1} = 0$, which is a contradiction. When $v_1 \ge 3$, $v_2 \ge v_1 \ge 3$. Therefore, $v_2 \ge 3$, and $v_i \ge i+1$ for $i = 1, 2$.

If $v_1 < v_2$, then obviously v_2 is not linearly dependent on v_1; if $v_1 = v_2$, then $v_1 = v_2 \ge 3$, and V_1 and V_2 form a system with rank 2. In either case, V_1, V_2 are linearly independent.

Suppose the claim is true for $i = n$ and consider $i = n+1$. Then we have either $v_n \ge n+2$ or $v_n = n+1$. If $v_n \ge n+2$, then $v_{n+1} \ge n+2$, $V_1, \ldots, V_n, V_{n+1}$ form a step-wise transpose Vandermonde-like system with rank $n+1$, and V_{n+1} is not a linear combination of $V_1, \ldots, V_n$. If $v_n = n+1$ and $v_{n+1} > n+1$, V_{n+1} is independent of $V_1, \ldots, V_n$.

If $v_n = v_{n+1} = n+1$, $V_1, \ldots, V_{n+1}$ form an $(n+1) \times (n+1)$ step-wise transpose Vandermonde-like system of rank $n+1$, which implies a solution $c_t = c_{t-1} = \cdots = c_{t-n} = 0$, a contradiction.

Now consider when the matrix $\mathcal{V}$ is t by t, V_t is independent from $V_1, \ldots, V_{t-1}$ and $\mathcal{V}$ is non-singular with solution $c_t = c_{t-1} = \cdots = c_1 = 0$, which is a contradiction. Therefore, $t > \kappa = d + 1 - r$. □

Consider a univariate polynomial f of degree d that is given in any power basis in which there are exactly r non-zero terms. If $r > (d+1)/2$, Theorem 1 provides a lower bound for the optimal sparsity of f as $d + 1 - r < \tau \le r$. In the case when $r \le (d+1)/2$, this is the sufficient condition for the unique sparsest shift (Lakshman and Saunders, 1996, Theorem 1), of which our Theorem 1 gives a simple proof:

Lemma 2. *For a univariate polynomial $f(x)$ with $\deg f(x) = d$, if there are exactly τ non-zero terms in the power basis of $(x+\theta)$ and $\tau \le (d+1)/2$, then θ is the unique sparsest shift of $f(x)$, which is an element in the quotient field of* D *(Lakshman and Saunders, 1996, Theorem 1). Again we assume that $\binom{d}{j} \ne 0$ in* D *for all $0 < j < d$.*

Proof. Suppose $s \ne \theta$, and there are t non-zero terms of f in the s-shifted basis, by Theorem 1, $t > d + 1 - \tau \ge (d+1)/2$. Now suppose that θ is an algebraic element over the quotient field of D. Then for a conjugate θ^* of θ we have $f(x) = \sum_i c_i(\theta)(x+\theta)^{e_i} = \sum_i c_i(\theta^*)(x+\theta^*)^{e_i}$, because $f(x) \in \mathsf{D}[x]$. □

Consider a multivariate polynomial $f(x_1, \ldots, x_n)$ and a multivariate sparsest shift $\theta = (\theta_1, \ldots, \theta_n)$. If there are m components θ_j of $\theta = (\theta_1, \ldots, \theta_n)$ so that each θ_j happens to be the sparsest shift of f in variable x_j, then each of those m components θ_j can be computed as a univariate shift of f in x_j, and the overall n-variate problem be brought down to an $(n-m)$-variate problem.

In the case the multivariate sparsest shift of f is very sparse, considering Lemma 2 on each variable in turn provides a sufficient condition for the uniqueness of the multivariate sparsest shift (see Lemma 3). Based on the fast algorithm for finding the unique sparsest rational shift in the univariate case (see Section 3.3), with high probability we can quickly determine whether such a shift exists, and obtain the shift if it does.

Lemma 3. *Let $\delta = \min_{1 \le j \le n}\{\deg_{x_j} f\}$. If f has exactly τ non-zero terms in the θ-shifted basis and that $\tau \le (\delta+1)/2$, then $\theta = (\theta_1, \ldots, \theta_n)$ is the unique sparsest shift of $f(x_1, \ldots, x_n)$, and θ_j is the unique sparsest shift of f in x_j for $1 \le j \le n$ (Grigoriev and Lakshman, 2000, cf. Lemma 2). Furthermore, $\theta \in \mathsf{K}^n$, where K is the field generated by the coefficients of f (Lakshman and Saunders, 1996, cf. Corollary 1).*

3. Finding sparsest shifts

Based on the early termination sparse interpolation algorithms (Kaltofen et al., 2000; Kaltofen and Lee, 2003), we present a class of algorithms for finding sparsest shifts: the interpolation steps are sensitive to the sparsity of the target polynomial in a given basis. We leave the shifts as variables in the procedure and solve the shift variables that minimize the interpolation steps. We will first concentrate on the case of power bases. Later in Section 3.4 we consider the Pochhammer and Chebyshev bases.

Section 2.1 showed how to form $g(w_1, \ldots, w_n) = f(x_1, \ldots, x_n)$ for a given polynomial f. All our algorithms can be employed for any given basis w_j. We shall focus on the standard power basis without losing generality.

Consider a polynomial $f \in \mathsf{D}[x_1, \ldots, x_n]$ represented in the s-shifted basis with $s = (s_1, \ldots, s_n)$ and $\overline{\mathsf{K}}$ the algebraic closure of D:

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{t} c_i (x_1+s_1)^{e_{i,1}} \cdots (x_n+s_n)^{e_{i,n}}, \qquad c_i \in \overline{\mathsf{K}}.$$

Note that t, c_i, and $e_{i,j}$ are all dependent on s. The problem of computing a sparsest shift within S is to find $s \in S$ such that t is minimized. Another notion is that of a T*-sparse shift* (within S), which is a point $s = (s_1, \ldots, s_n) \in S$ such that for the number of shifted terms we have $t \le T$. Algorithms for computing all T-sparse shifts take T as an additional input.

We introduce n indeterminates $z_1, \ldots, z_n$ to serve as shift variables, and manipulate $f(x_1, \ldots, x_n)$ in the symbolic $(z_1, \ldots, z_n)$-shifted basis $y_j = x_j + z_j$ for $1 \le j \le n$. Then by applying (4), with $W = [y_1, \ldots, y_n]^{\mathrm{Tr}}$, $A_0 = [z_1, \ldots, z_n]^{\mathrm{Tr}}$, and $A = I_n$, we obtain $f(y_1 - z_1, \ldots, y_n - z_n)$. Now, consider the interpolation of f in the basis of y_j symbolically by the early termination Ben-Or/Tiwari algorithm (Kaltofen et al., 2000; Kaltofen and Lee, 2003): with $\alpha_i = f(y_1^i - z_1, \ldots, y_n^i - z_n)$ the Berlekamp/Massey algorithm is carried out on a sequence of polynomials $\{\alpha_i\}_{i \geq 1}$, and the discrepancies become rational functions.

To avoid the GCD operations on the arising numerators and denominators of such rational functions, we can implement the fraction-free Berlekamp/Massey algorithm (Giesbrecht et al., 2002, Section 2) for computing the discrepancies Δ_i. Therefore, $\{\alpha_i\}_{i \geq 1}$ is a sequence of polynomials in $\overline{\mathsf{K}}[z_1, \ldots, z_n][y_1, \ldots, y_n]$, and the discrepancies Δ_i are polynomials in $y_1, \ldots, y_n$ over $\overline{\mathsf{K}}[z_1, \ldots, z_n]$. The following lemma is based on the early termination (Kaltofen et al., 2000; Kaltofen and Lee, 2003) of the Ben-Or/Tiwari algorithm.

Lemma 4. *When a shift $s = (s_1, \ldots, s_n) \in \overline{\mathsf{K}}^n$ is given, the discrepancies Δ_i evaluated at $(z_1, \ldots, z_n) = (s_1, \ldots, s_n)$ are non-zero polynomials in y_i for $1 \leq i \leq 2t$, and zero polynomials for all $i \geq 2t + 1$, where t is the number of terms of the target polynomial f in the s-shifted basis.*

All our algorithms manipulate the discrepancies Δ_i, and we present our algorithms in three categories. The *symbolic algorithms* of Section 3.1 treat Δ_i as polynomials in $\overline{\mathsf{K}}[z_1, \ldots, z_n][y_1, \ldots, y_n]$ and work in deterministic polynomial time for constant n over any field over which algebraic systems can be solved. The *single projection algorithms* in Section 3.2 evaluate each y_j at a value p_j to increase efficiency. Finally, in Section 3.3, we present the *double projection algorithm* for polynomials $f \in \mathbb{Q}[x]$, wherein the Δ_i are evaluated at random $y = p \in \mathbb{Z}$ as well as random shifts $z = s \in \mathbb{Z}$. This yields a particularly efficient algorithm for rational polynomials.

3.1. *Symbolic algorithms*

Our symbolic algorithms are all deterministic, they treat both the shifted basis y_i and the shift variables z_i as indeterminates. Consider the fraction-free Berlekamp/Massey algorithm processing the sequence $\{\alpha_i\}_{i \geq 1}$ with $\alpha_i = f(y_1^i - z_1, \ldots, y_n^i - z_n)$: the discrepancies Δ_i are polynomials in $y_1, \ldots, y_n$ over $\overline{\mathsf{K}}[z_1, \ldots, z_n]$. Based on Lemma 4, we seek sparsest shifts for f within S by solving for $z \in S$ that minimize i such that $\Delta_{i=2t+1}$ is a zero polynomial in $\overline{\mathsf{K}}[y_1, \ldots, y_n]$.

Algorithm. `MultiSparsestShifts <symbolic>`

```
Input:  ▸ $f(x_1, \ldots, x_n) \in \mathsf{D}[x_1, \ldots, x_n]$: the input polynomial;
        ▸ $S \subset \overline{\mathsf{K}}^n$: $S \neq \varnothing$, the shifts are constrained within $S$.
Output: ▸ $\theta \in S$: the sparsest shifts for $f$ in $S$.
(1) [Compute $\Delta_i$].
    Perform the fraction-free Berlekamp/Massey algorithm on $\{\alpha_i\}_{i \geq 1}$;
(2) [Solve for the first $\Delta_{i=2t+1} = 0$, the zero polynomial in $\overline{\mathsf{K}}[y_1, \ldots, y_n]$].
        If $i = 2t + 1$, an odd integer, then
            if there is $(\theta_1, \ldots, \theta_n) \in S$ such that $\Delta_i(\theta_1, \ldots, \theta_n, y_1, \ldots, y_n)$ is a zero
            polynomial in $\overline{\mathsf{K}}[y_1, \ldots, y_n]$, then
                Break out of the loop;
                Return all the solutions $\theta \in S$ for $\Delta_i = 0 \in \overline{\mathsf{K}}[y_1, \ldots, y_n]$.
```

The algorithm always terminates: for any $s \in S \neq \emptyset$, $\varDelta_{2t+1} = 0$ when t is the number of terms of f in the s-shifted basis. In step (2), a discrepancy $\varDelta_i = \sum_{j=1}^{\kappa} g_{i,j} \cdot y_1^{\sigma_{j,1}} \cdots y_n^{\sigma_{j,n}}$ becomes a zero polynomial in $\overline{\mathsf{K}}[y_1, \ldots, y_n]$ if the system of polynomial equations

$$\begin{aligned} g_{i,1}(z_1, \ldots, z_n) &= 0 \\ &\vdots \\ g_{i,\kappa}(z_1, \ldots, z_n) &= 0 \end{aligned}$$

has a solution in S, and the problem is thus reduced to solving an algebraic system. We add that for multivariate polynomials, transcendental shifts are possible, for instance $x_1 + x_2 - 1 = (x_1 + \vartheta) + (x_2 - \vartheta - 1)$. In this case the variety of shift points is of dimension higher than 0.

For the problem of computing T-sparse shifts, we note that in Lemma 4, a zero of $\varDelta_k$ stays a zero of $\varDelta_i$ for all $i \geq k$, and the shifts that make $\varDelta_{2T+1}$ a zero polynomial (in y_j) are all s that make $t \leq T$. We find T-sparse shifts within S by solving all $s \in S$ such that $\varDelta_{2T+1}(s_1, \ldots, s_n, y_1, \ldots, y_n) = 0$.

When the polynomial $f(x)$ is univariate, a number of special "tricks" can be employed. Under the z-shifted basis $y = x + z$, the discrepancies $\varDelta_i$ are polynomials in y with coefficients in $\overline{\mathsf{K}}[z]$, and every $\varDelta_i$ is a product of its primitive part $\varphi_i(y) \in \overline{\mathsf{K}}[z][y]$ and content $g_i \in \overline{\mathsf{K}}[z]$:

$$\varDelta_i = g_i \cdot \varphi_i(y). \tag{7}$$

A sparsest shift $\theta \in S$ occurs at the first i such that $\varDelta_i$ becomes a zero polynomial in y, that is, when $g_i = 0$. If $S = \overline{\mathsf{K}}$, at the first time g_i is a non-trivial polynomial in $\overline{\mathsf{K}}[z]$, there is a solution to $g_i = 0$ and the solutions are the sparsest shifts for f. Since all zeros of g_i stay zeros of g_{i+1}, we can just look for the first non-trivial GCD of g_i and g_{i+1} in $\overline{\mathsf{K}}[z]$.

Algorithm. `UniSparsestShifts <symbolic>`

```
Input:   ► $f(x) \in \mathsf{D}[x]$: a univariate polynomial;
Output:  ► $\theta \in \overline{\mathsf{K}}$: the sparsest shifts for $f$ in $\overline{\mathsf{K}}$.
(1) [Compute $\varDelta_i$].
    Perform the fraction-free Berlekamp/Massey algorithm on $\{\alpha_i = f(y^i - z)\}_{i \geq 1}$;
(2) [Compute $\gcd(g_{i-1}, g_i)$, the content of $\gcd(\varDelta_{i-1}, \varDelta_i)$].
        If $i = 2t + 2$, an even integer, then
            if $\gcd(g_{i-1}, g_i)$ is non-trivial in $\overline{\mathsf{K}}[z]$, then
                Break out of the loop;
                Return all the solutions of $g_{i-1}(z) = 0$ in $\overline{\mathsf{K}}$.
```

This algorithm requires a root finder in $\overline{\mathsf{K}}[z]$. Likewise, to find all T-sparse shifts for a univariate polynomial, we solve $\gcd(g_{2T+1}, g_{2T+2}) = 0$.

We can easily determine the complexity of this algorithm in terms of operations in K (not including the cost of the root finding). We first observe that the cost of running the Berlekamp/Massey algorithm in step (1) dominates other costs. Also, the degrees of the polynomials involved in the computation do not get larger than $O(\tau^2 d)$ in y and $O(\tau d)$ in z (assuming the input f has degree d). Thus, the total cost is $O(\tau^2 \mathsf{M}(\tau^3 d^2))$ operations in K, where $\mathsf{M}(m)$ is the cost of multiplying two univariate polynomials of degree m. $\mathsf{M}(m) = O(m^2)$ using standard polynomial arithmetic and $\mathsf{M}(m) = O(m \log m \log \log m)$ using asymptotically fast polynomial arithmetic. Theoretically there is an asymptotically faster alternative by replacing the Berlekamp/Massey algorithm with the algorithm of Brent et al. (1980).

3.2. Single projection algorithms

The efficiency of symbolic algorithms can be improved substantially by projecting variables y_j to values p_j. For simplicity, we describe projection algorithms as finding sparsest shifts within certain algebraic extensions. However, they can all be modified as being restricted to a non-empty subset S.

Now consider the discrepancy from the previous subsection, $\Delta_{2T+1}(z_1, \dots, z_n, y_1, \dots, y_n)$, evaluated at $(y_1, \dots, y_n) = (p_1, \dots, p_n)$, where p_j are distinct values.

Algorithm. `MultiSparseShiftsEquation <one proj>`

```
Input:  ▸ $f(x_1, \dots, x_n) \in \mathsf{D}[x_1, \dots, x_n]$: the input polynomial;
        ▸ $T$: a positive integer; $T$-sparse shifts for $f$ are being considered.
Output: ▸ $\Delta_{2T+1}$: a polynomial; $T$-sparse shifts of $f$ have to satisfy $\Delta_{2T+1} = 0$.
(1) [Choose the projection values].
    Pick distinct random values $p_1, \dots, p_n$;
(2) [Compute $\Delta_{2T+1}$].
    Carry out the fraction-free Berlekamp/Massey algorithm on $\{\alpha_i\}_{1 \le i \le 2T+1}$ with $\alpha_i = f(p_1^i - z_1, \dots, p_n^i - z_n)$.
```

The output polynomial equation might contain roots that are not T-sparse shifts of f, but if we restrict the shifts within a set S, the single constraint $\Delta_{2T+1} = 0$ may be sufficient to locate all T-sparse shifts within S. Additional equations can be generated by running the algorithm for different random p_j's. Eventually all false solutions, the zeros that do not yield a T-sparse shift, will be eliminated from a system of polynomial equations with enough distinct p_j's.

Theorem 5. *Consider a system of polynomial equations such that each equation $\Delta_{i,2T+1} = 0$ is an output of algorithm* `MultiSparseShiftsEquation` *that projects $(y_1, \dots, y_n)$ to $q_i = (q_{i,1}, \dots, q_{i,n})$. If there are enough equations $\Delta_{i,2T+1} = 0$ with distinct q_i, then all the solutions are T-sparse shifts of f.*

Proof. Consider the symbolic discrepancy: $\Delta_{2T+1} = \sum_{j=1}^{\kappa} g_j \cdot y_1^{\sigma_{j,1}} \cdots y_n^{\sigma_{j,n}} = \sum_{j=1}^{\kappa} g_j \cdot y^{\sigma_j}$, with $y^{\sigma_j} = y_1^{\sigma_{j,1}} \cdots y_n^{\sigma_{j,n}}$ and $g_j \in \overline{\mathsf{K}}[z_1, \dots, z_n]$. The solutions to $g_1 = \cdots = g_\kappa = 0$ are T-sparse shifts of f. Now let $q_i^{\sigma_j} = q_{i,1}^{\sigma_{j,1}} \cdots q_{i,n}^{\sigma_{j,n}}$, the projection of y^{σ_j} at $q_i = (q_{i,1}, \dots, q_{i,n})$, and consider the following system:

$$\begin{bmatrix} q_1^{\sigma_1} & q_1^{\sigma_2} & \cdots & q_1^{\sigma_\kappa} \\ q_2^{\sigma_1} & q_2^{\sigma_2} & \cdots & q_2^{\sigma_\kappa} \\ \vdots & \vdots & \ddots & \vdots \\ q_\kappa^{\sigma_1} & q_\kappa^{\sigma_2} & \cdots & q_\kappa^{\sigma_\kappa} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_\kappa \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{8}$$

Given enough distinct q_i, we eventually obtain a non-singular system in (8), which provides solutions to $g_1 = \cdots = g_\kappa = 0$. These are only solutions to the initial system of equations. □

In the univariate case, with high probability all the false solutions can be eliminated by projecting y to two different random values. Consider $\Delta_i(y) = g_i \cdot \varphi_i(y)$ in (7) and distinct random values p, q. By the Schwartz–Zippel lemma (Schwartz, 1980), $\gcd(\Delta_i(p), \Delta_i(q)) = \gcd(g_i \cdot \varphi_i(p), g_i \cdot \varphi_i(q)) = g_i$ with high probability and our next algorithm follows.

Algorithm. `UniSparsestShifts <one proj, two seq>`

```
Input:  ► $f(x) \in \mathsf{D}[x]$: a univariate polynomial.
Output: ► $\theta$: the sparsest shifts for $f$ with high probability.
(1) [Choose the projection values $p$ and $q$].
    Pick distinct random values $p, q$;
(2) [Compute $\Delta_i(p)$ and $\Delta_i(q)$].
    Perform the fraction-free Berlekamp/Massey algorithm on $\{\alpha_i = f(p^i - z)\}_{i\geq 1}$ and
    $\{\beta_i = f(q^i - z)\}_{i\geq 1}$;
(3) [Compute $\gcd(\Delta_i(p), \Delta_i(q))$].
        If $i = 2t + 1$, an odd integer, then
            if $\gcd(\Delta_i(p), \Delta_i(q)) = g(z)$ is non-trivial in $\overline{\mathsf{K}}[z]$, then
                Break out of the loop;
                Return all the solutions of $g(z) = 0$ in $\overline{\mathsf{K}}$.
```

To further increase the probability of correctness, we can project y to more distinct random values $q_1, \ldots, q_k$ and form a projection sequence for each of them. Then we look for the first $i = 2t + 1$ such that $\gcd(\Delta_i(q_1), \ldots, \Delta_i(q_k)) = g$ is non-trivial in $\mathsf{K}[z]$.

Much as in the case of the `UniSparsestShifts <symbolic>` algorithm above, we can determine the complexity of this algorithm in terms of operations in K. Again, the cost of running the Berlekamp/Massey algorithm in step (2) dominates. The degrees of the polynomials involved in the computation do not get larger than $O(td)$, where $d = \deg f$, and all polynomials are univariate in z after the projections in step (1). Thus, the total cost is $O(\tau^2 \mathsf{M}(\tau d))$ operations in K.

We can even reduce the projection to a single sequence by taking GCD's of subsequent elements in the sequence. Recall the primitive part of Δ_i in (7), we need to make sure there is no non-trivial GCD of $\varphi_i(p), \varphi_{i+1}(p), \ldots$ in $\mathsf{K}[z]$ for all p.

Theorem 6. *Suppose that the sparsest shift of $f(x)$ in $\overline{\mathsf{K}}$ has $\tau < \deg(f) + 1$ terms and assume that $\binom{\deg(f)}{j} \neq 0$ for all $0 < j < \deg(f)$ when computed as an element in D. Then for $\Gamma = \mathrm{GCD}_{2\tau+1\leq i\leq 2\deg(f)+1}(\Delta_i(z, y))$ (over the quotient field of D) we have $\Gamma = g(z)\gamma(y)$ where $g(z) \in \mathsf{D}[z]$ and $\gamma(y) \in \mathsf{D}[y]$.*

Proof. As stated above, if $\Gamma(\theta, y) = 0$ for some θ in the algebraic closure of the quotient field of D, denoted by $\overline{\mathsf{K}}$, then $f(y - \theta)$ is τ-sparse in y. By assumption, there exists such a shift, and therefore $z - \theta$ divides Γ. As in (7) we factor $\Gamma(z, y) = g(z)\gamma(z, y)$, where $g \in \mathsf{D}[z]$ and $\gamma(z, y) \in \mathsf{D}[z, y]$ whose content in $\mathsf{D}[z]$ is 1. We claim that $\gamma(z, y) \in \mathsf{D}[y]$.

Let us suppose the contrary. Then there exists an element σ in the algebraic closure of $\mathsf{D}(z)$ and transcendental over $\overline{\mathsf{K}}$ such that $\gamma(z,\sigma) = 0$. We thus have that $\Delta_i(z,\sigma) = 0$ for all $2\tau + 1 \le i \le 2\deg(f) + 1$. Since the terms σ^i are all distinct, we then get from the Ben-Or/Tiwari algorithm, using $p = \sigma$ and re-interpreting the coefficient field of f to be the algebraic closure of $\mathsf{D}(z)$, that $f(y - z)$ is τ-sparse. Let $d = \deg(f)$ and $c_d \ne 0$ be the leading coefficient of f. However, the term $c_d\binom{d}{j}z^{d-j}$ is unique in the coefficient of y^j of $f(y - z)$, so $f(y - z)$ has actually $d + 1$ non-zero terms over $\mathsf{D}[z]$. □

The algorithm using a single projected sequence is as follows:

Algorithm. `UniSparsestShifts <one proj, one seq>`

```
Input:  ▶ $f(x) \in \mathsf{D}[x]$: a univariate polynomial;
        ▶ $\delta$: an upper bound on deg $f$.
Output: ▶ $\theta$: the sparsest shifts for $f$ in $\overline{\mathsf{K}}$ with high probability.
(1) [Choose a projection value].
    Pick a random value $p$;
(2) [Compute $\Delta_1, \ldots, \Delta_{2\delta+1}$].
    Compute $\Delta_1, \ldots, \Delta_{2\delta+1}$ by the fraction-free Berlekamp/Massey algorithm on
    $\{\alpha_i\}_{1\le i\le 2\delta+1}$ with $\alpha_i = f(p^i - z)$;
(3) [Minimize t so that gcd($\Delta_{2t+1}, \ldots, \Delta_{2\delta+1}$) is non-trivial].
    For $t = \delta, \delta - 1, \delta - 2, \ldots$ do
        if gcd($\Delta_{2\delta+1}, \ldots, \Delta_{2t}$) becomes trivial in $\mathsf{K}[z]$, then
            Break out of the loop;
    Return all solutions of gcd($\Delta_{2t+1}, \ldots, \Delta_{2\delta}$) = 0 in $\overline{\mathsf{K}}$.
```

We conjecture that instead of taking the GCD of all discrepancies up to $2\deg f + 1$, we can only look for the GCD of a much smaller number of discrepancies after the sparsest case $\Delta_{2\tau+1}$ is reached.

3.3. Two projections: finding the sparsest shifts of a rational polynomial

When $f \in \mathbb{Q}[x]$, we can project the sequence $\{f(y^i - z)\}_{i\ge 1}$ both on a random y and random z from $\mathbb{Z}$, and use the multiplicative structure of the integers to recover the sparsest shift. Thus, finding the sparsest shift will be reduced to running the Berlekamp/Massey algorithm on a small number of integer sequences. The existence of a large prime factor in the GCD's of two discrepancies will reveal the sparsest shift. This improves the efficiency. It also allows us to work completely with a black-box representation for f, requiring only the value of f at points in $\mathbb{Z}$.

Finding factors of a black-box polynomial

We begin by demonstrating a general algorithm for finding a linear factor in one variable of a black-box bivariate polynomial. This will be applied to the discrepancy polynomials.

Let $\Phi \in \mathbb{Q}[z, y]$ be a black-box polynomial of degree C in y and degree d in z. Suppose that

$$\Phi(z, y) = (az - b)^e\,\Psi(z, y),$$

where $a, b \in \mathbb{Z}$ are relatively prime, $e \geq 1$, and $\Psi(z, y) \in \mathbb{Q}[x, z]$ has no non-trivial factor in $\mathbb{Z}[z]$. In this section we give a Monte Carlo algorithm to find a and b with a small constant number of evaluations of Φ.

A number $m \in \mathbb{Q}$ is said to be μ-smooth, for some $\mu > 1$, if all prime factors of both the numerator and denominator of m are less than μ. A polynomial $\Psi \in \mathbb{Z}[z, y]$ is primitive if the GCD of all its coefficients is one. A polynomial $\Phi \in \mathbb{Q}[z, y]$ is μ-primitive if it is a μ-smooth number times a primitive, integer polynomial. For any $\Psi = \sum_{ij} \Psi_{ij} y^i z^j \in \mathbb{Z}[z, y]$, let $\| \Psi \| = \max | \Psi_{ij} |$. The *height* of a rational number $\alpha/\beta \in \mathbb{Q}$ (where $\gcd(\alpha, \beta) = 1$) is $\mathcal{H}(\alpha/\beta) = \max\{|\alpha|, |\beta|\}$. Define the *denominator* $\text{denom}(\Phi)$ of $\Phi \in \mathbb{Q}[z, y]$ as the LCM of the denominators of its coefficients. The *content* of Φ is then defined as the usual content of the integer polynomial $\text{denom}(\Phi) \cdot \Phi$. The height of $\Phi \in \mathbb{Q}[z, y]$ is $\mathcal{H}(\Phi) = \max\{|\text{denom}(\Phi)|, \|\text{denom}(\Phi) \cdot \Phi\|\}$. Note that this is the height of Φ in the standard, unshifted, power basis.

To begin with we will insist that Φ is *μ-primitive*, and treat the general case separately below.

Algorithm. `FindLinFac`

```
Input:  ▸ Black box for $\Phi \in \mathbb{Q}[z, y]$;
        ▸ Bounds $C \geq \deg_y \Phi$, $D \geq \deg_z \Phi$, $H > \mathcal{H}(\Phi)$;
        ▸ $S >$ height of the sought linear factor;
        ▸ a smoothness bound $\mu$;
          $\Phi$ is assumed to be $\mu$-primitive, and
          $\mu \geq \max\{17, S^2, 11C\sqrt{D}(H + 2C + 2),$
                  $540CD^2 \log H \log(C + D + \log H)\}$;
Output: ▸ A candidate factor $az - b$ of $\Phi$, where $a, b \in \mathbb{Z}$ are relatively prime;
          or a report "No linear factor in $z$ exists";

 (1) $\mathcal{L} = \{0, \ldots, \mu^2 - 1\}$;
 (2) Choose random $\gamma_1, \gamma_2, \sigma \in \mathcal{L}$;
 (3) Let $\overline{q} = \gcd(\text{numer}(\Phi(\sigma, \gamma_1)), \text{numer}(\Phi(\sigma, \gamma_2)))$;
 (4) Let $q = \overline{q}/m$, with $m$ the largest $\mu$-smooth factor of $\overline{q}$;
 (5) If $q = 1$
 (6)     Then Return "No linear factor in $z$ exists";
 (7)     Else
 (8)     Find $w$ and largest $e \geq 1$ such that $q = w^e$
 (9)     If $w < 2S^2$
(10)         Then Return "Failure";
(11)         Else Return $a, b \in \mathbb{Z}$ such that
             $\gcd(a, b) = 1$, $|a|, |b| \leq S$, and $-b/a \equiv \sigma \bmod w$;
```

Theorem 7. *For any black box $\Phi \in \mathbb{Z}[z, y]$ meeting the input criteria,* `FindLinFac` *works correctly as stated with probability at least* $1/5$ *on any invocation.*

Comments

- While the algorithm is defined for μ-primitive polynomials in $\mathbb{Z}[z, y]$, the reader is encouraged to think of these as simply primitive polynomials in $\mathbb{Z}[z, y]$. The μ-smooth content is the only rational (that is, non-integer) part of the computation.
- The input S, the height of the desired linear factor, can be replaced with $\|\Phi\|$. However, if we have a priori knowledge of a smaller factor (as we do later in this section), this input may be useful.
- The algorithm can be run repeatedly until a factor is found, or the user is satisfied that with sufficiently high probability no linear factor exists.
- The probability of success is undoubtedly much higher than is proven here.
- If w is too small and the algorithm reports "failure" in step (10), we get a useful modular relation between a and b. Collecting these may allow us to construct a, b without ever getting a really large w.

To prove Theorem 7, we require a number of lemmas. The first says that if we project a primitive bivariate polynomial randomly along one coordinate twice, then we expect to get two relatively prime univariate polynomials whose contents do not share any large prime factors.

Lemma 8. *Let* $\Psi \in \mathbb{Z}[z, y]$ *be primitive with* $\deg_y \Psi \le C$ *and* $\deg_z \Psi \le D$. *Let* $\mu \ge 11CD^{1/2}(\log\|\Psi\| + 2C + 2)$. *For* γ_1, γ_2 *chosen randomly from* $\{0, \ldots, \mu^2 - 1\}$, *the probability that* $\gcd(\Psi(z, \gamma_1), \Psi(z, \gamma_2)) = 1$, *and that no prime* $\ge\mu$ *divides the contents of both* $\Psi(z, \gamma_1)$ *and* $\Psi(z, \gamma_2)$, *is at least* $9/10$.

Proof. We first show that for randomly chosen $\gamma_1, \gamma_2 \in \{0, \ldots, \mu^2 - 1\}$, the resultant r of $\Psi(z, \gamma_1)$, and $\Psi(z, \gamma_2)$ is non-zero with high probability. This will imply $\Psi(z, \gamma_1)$ and $\Psi(z, \gamma_2)$ are relatively prime. Let y_1, y_2 be two new indeterminates and consider the resultant $R(y_1, y_2) \in \mathbb{Z}[y_1, y_2]$ of $\Psi(z, y_1)$ and $\Psi(z, y_2)$ as polynomials in $\mathbb{Q}(y_1, y_2)[z]$. R has degree at most $2CD$. For randomly chosen $\gamma_1, \gamma_2 \in \{0, \ldots, \mu^2 - 1\}$, $R(\gamma_1, \gamma_2) \neq 0$ with probability at least $1 - 2CD/\mu^2$ by the Schwartz–Zippel lemma.

Now write $\Psi(z, y) = \sum_{0\le i\le D} \Psi_i(y)z^i$. For any $\gamma \in \{0, \ldots, \mu^2 - 1\}$, for all terms Ψ_j, we have $|\Psi_j(\gamma)| \le \|\Psi\| \cdot \mu^{2C+2}$. Assume that $\gcd(\Psi(z, \gamma_1), \Psi(z, \gamma_2)) = 1$, so in particular, $\Psi(z, \gamma_1) \neq 0$. The content of $\Psi(z, \gamma_1)$ is at most $\|\Psi\| \cdot \mu^{2C+2}$, and this has at most $\log_\mu(\|\Psi\| \cdot \mu^{2C+2}) \le \log\|\Psi\| + 2C + 2$ prime factors $\ge\mu$. Since Ψ is primitive, for each prime p dividing the content of $\Psi(z, \gamma_1)$ there exists an i such that $\Psi_i(y) \not\equiv (0) \bmod p$. There are at most C integers $\gamma_2 \in \{0, \ldots, p-1\}$ such that $\Psi_i(\gamma_2) \equiv 0 \bmod p$. For any prime $p \ge \mu$, there are at most $C\mu$ integers $\gamma_2 \in \{0, \ldots, \mu^2 - 1\}$ such that $\Psi_i(\gamma_2) \equiv 0 \bmod p$. The total number of $\gamma \in \{0, \ldots, \mu^2 - 1\}$ such that there exists a $p \ge \mu$ dividing the contents of both $\Psi(z, \gamma_1)$ and $\Psi(z, \gamma_2)$ is then at most $C(\log\|\Psi\| + 2C + 2)\mu$.

The probability that either $\gcd(\Psi(z, \gamma_1), \Psi(z, \gamma_2)) \neq 1$ or there is a prime $p \ge \mu$ which divides both their contents, is at most $2CD/\mu^2 + C(\log\|\Psi\| + 2C + 2)/\mu < 1/10$, by our choice of μ. □

The next lemma simply says that the GCD of an evaluation of two relatively prime integer polynomials is generally smooth.

Lemma 9. *Let $h_1, h_2 \in \mathbb{Z}[y]$ be relatively prime, primitive polynomials of degree $d \leq D$ and resultant $r \in \mathbb{Z}$, where $\mu \geq 10D\log r$. For a randomly chosen $\sigma \in \{0, \ldots, \mu^2 - 1\}$, $\gcd(h_1(\sigma), h_2(\sigma))$ is μ-smooth with probability at least $9/10$.*

Proof. Since h_1, h_2 are relatively prime, there exist $u_1, u_2 \in \mathbb{Z}[y]$ such that $u_1(y)h_1(y) + u_2(y)h_2(y) = r$. Thus, if any prime divides $h_1(\sigma)$ and $h_2(\sigma)$, that prime divides r as well. Suppose then that p is a prime dividing r. Then there exists $u_1^{(p)}, u_2^{(p)}, w^{(p)} \in \mathbb{Z}[y]$ such that $w^{(p)}$ is the GCD of h_1, h_2 modulo p, and $0 < \deg w^{(p)} < d$, and

$$u_1^{(p)}(y)h_1(y) + u_2^{(p)}(y)h_2(y) = w^{(p)}(y) + pQ^{(p)}(y)$$

for some $Q^{(p)} \in \mathbb{Z}[y]$. If p divides $h_1(\sigma)$ and $h_2(\sigma)$, we have $w^{(p)}(\sigma) \equiv 0 \bmod p$. For any prime p, the number of $\sigma \in \{0, \ldots, p - 1\}$ such that $w^{(p)}(\sigma) \equiv 0 \bmod p$ is less than D. For primes $p \geq \mu$ the number of $\sigma \in \{0, \ldots, \mu^2 - 1\}$ such that $w^{(p)} \equiv 0 \bmod p$ is less than $D\mu$.

We know r has at most $\log r$ prime factors, so the probability that $w^{(p)}(\sigma) \equiv 0 \bmod p$ for *any* prime $p > \mu$ is at most $D\log(r)/\mu < 1/10$ by our choice of μ. □

We look now at the probability that a number in an arithmetic progression is *rough*, i.e. has a large prime factor. This theorem is an extension of an exercise of Knuth (1983). Let $a, b \in \mathbb{Z}$ be relatively prime. We say that an integer $x \in \{0, \ldots, \mu^2 - 1\}$ is $(\mu; a, b)$-rough if the largest prime factor of $ax + b$ is greater than μ.

Lemma 10. *Let $a, b \in \mathbb{Z}$ be relatively prime and $\mu \geq \max\{a^2, b, 17\}$. The number of $(\mu; a, b)$-rough integers x with $0 \leq x < \mu^2$ is at least $\mu^2/4$.*

Proof. We assume $a > 0$. For a prime $p > \mu\sqrt{a}$, there is a unique x_0 such that $0 \leq x_0 < p$ and $ax_0 + b \equiv 0 \bmod p$. Thus, the sequence of all x $(0 \leq x < \mu^2)$ such that $ax + b \equiv 0 \bmod p$ is $x_0, x_0 + p, \ldots, x_0 + kp$, where $x_0 + kp < \mu^2$ and $x_0 + (k + 1)p \geq \mu^2$. For any p there are at least $\mu^2/p - 1$ such numbers.

Any number can appear in the sequence for at most one prime. To see this, assume that x appears in the sequences for distinct primes p and q, with $\mu\sqrt{a} < p < q < \mu^2$. Then $pq \mid ax + b$. Since $q > \mu\sqrt{a} + 2$, $pq > \mu^2 a + 2\mu\sqrt{a}$. But $ax + b$ is at most $a\mu^2 + b < a\mu^2 + \mu < a\mu^2 + 2\mu\sqrt{a} < pq$.

Summing all primes p such that $\mu\sqrt{a} < p < \mu^2$ (and using the fact that $\sqrt{\alpha} \leq \mu^{1/4}$), we count

$$\begin{aligned}
\sum_{\mu\sqrt{a}<p<\mu^2} \frac{\mu^2}{p} - 1 &= \mu^2 \sum_{\mu\sqrt{a}<p<\mu^2} \frac{1}{p} - \sum_{\mu\sqrt{a}<p<\mu^2} 1 \\
&\geq \mu^2 \left(\log\log\mu^2 - \log\log\mu\sqrt{a} - \frac{1}{2\log^2(\mu^2)} - \frac{1}{2\log^2(\mu\sqrt{a})} \right) - \pi(\mu^2) \\
&\geq \mu^2 \left(\log\log\mu^2 - \log\log\mu^{5/4} - \frac{1/8}{\log^2\mu} - \frac{8/25}{\log^2\mu} \right) - \pi(\mu^2) \\
&\geq \mu^2 \left(\log\frac{8}{5} - \frac{89/200}{\log^2\mu} \right) - \frac{\mu^2}{-1.5 + \log\mu^2},
\end{aligned}$$

which is $\geq \mu^2/4$ for $\mu \geq 33$. Here $\pi(m)$ is the number of primes less than or equal to m, and Theorem 2 of Rosser and Schoenfeld (1962) shows $\pi(m) < m/(-1.5 + \log(m))$ for $m > 5$. We also use Theorems 5 and 6 from Rosser and Schoenfeld (1962) which show that

$$\log\log m + B - \frac{1}{2\log^2 m} < \sum_{p \leq m} \frac{1}{p} < \log\log m + B + \frac{1}{2\log^2 m}$$

for $m \geq 286$. We verify the theorem for all $\mu \geq 17$. □

Proof of Theorem 7. Start by considering a primitive polynomial $\Psi \in \mathbb{Z}[z, y]$ of degree $c \leq C$ in y and $d \leq D$ in z, that has no non-trivial factor in z alone.

By Lemma 8, for randomly chosen $\gamma_1, \gamma_2 \in \mathcal{L}$, with probability at least $9/10$, $P_1 := \Psi(z, \gamma_1)$ and $P_2 := \Psi(z, \gamma_2)$ are relatively prime, and their contents do not share any prime factor $\geq \mu$. Assume this is indeed the case for our choice of γ_1, γ_2.

It is easily derived that $\| \Psi(z, \gamma_i)\| \leq (102C^2D(\log\|\Psi\| + 2C + 2))^{C+1} \cdot \|\Psi\|$. Thus, the resultant r of $\Psi(z, \gamma_1)$ and $\Psi(z, \gamma_2)$ is at most $(2D)^{2D} \cdot (102C^2D \cdot (\log\|\Psi\| + 2C + 2)^2)^{2D(C+1)} \cdot \|\Psi\|^{2D}$. Simplifying this, we find that $\log r \leq 54CD\log\|\Psi\| \cdot \log(C + D + \log H)$. By Lemma 9, for a randomly chosen $\sigma \in \mathcal{L}$, $\gcd(\Psi(\sigma, \gamma_1), \Psi(\sigma, \gamma_2))$ is μ-smooth with probability at least $9/10$.

Write $P_1(z) = c_1 \cdot h_1(z)$ and $P_2(z) = c_2 \cdot h_2(z)$, where $c_1, c_2 \in \mathbb{Z}$ are the contents of P_1, P_2 respectively, and $h_1, h_2 \in \mathbb{Z}[z]$ are primitive. For a random $\sigma \in \mathcal{L}$, we compute $G = \gcd(\Psi(\sigma, \gamma_1), \Psi(\sigma, \gamma_2)) = \gcd(c_1, c_2) \cdot \gcd(h_1(\sigma), h_2(\sigma))$. By Lemma 8, $\gcd(c_1, c_2)$ is μ-smooth. By Lemma 9, $\gcd(h_1(\sigma), h_2(\sigma))$ is μ-smooth with probability at least $9/10$. Thus G is μ-smooth with probability at least $81/100$.

Now consider the full case when $\Phi(z, y) = m \cdot (az + b)^e \cdot \Psi(z, y)$, where $m \in \mathbb{Q}$ is μ-smooth, $a, b \in \mathbb{Z}$ are relatively prime, and Ψ is primitive and has no factor purely in $\mathbb{Z}[z]$. Then $\overline{q} = m \cdot (a\sigma - b)^e \cdot \gcd(\Psi(\sigma, \gamma_1), \Psi(\sigma, \gamma_2))$. From above we see $\gcd(\Psi(\sigma, \gamma_1), \Psi(\sigma, \gamma_2))$ is μ-smooth with probability at least $81/100$. Thus w is equal to the factor of $a\sigma - b$ which is not μ-smooth. Both $|a|$ and $|b|$ are less than S. By Lemma 10, $(a\sigma + b)$ has a prime factor of size greater than μ with probability at least $1/4$, and in this case we recover a, b as described in step (11). To conclude, for any input, on any invocation the algorithm succeeds with probability at least $(81/100) \cdot (1/4) \geq 1/5$. □

Approximating the denominator and content

To complete the general algorithm, we must identify the μ-primitive part of a black-box polynomial. The following algorithm does this with two evaluations of the black box.

Algorithm. `DenomAndCont`

```
Input:   ▶ Black box for $f \in \mathbb{Q}[y]$;
         ▶ $D \geq \deg f$, $H \geq \mathrm{height}(f)$;
         ▶ a desired smoothness bound $\mu \geq 4D(\log H + 2D + 2)$;
Output:  ▶ a candidate $\omega \in \mathbb{Q}$ such that content of $\omega f$ is $\mu$-smooth;

(1) Let $\mathcal{L} = \{0, \ldots, \mu^2 - 1\}$;
(2) Choose a random $\alpha_0 \in \mathcal{L}$ and compute $\nu_0 = f(\alpha_0) \in \mathbb{Q}$;
    If $\nu_0 = 0$ the goto (2);
```

```
(3) Choose random $\alpha_1 \in \mathcal{L}$; compute $\nu_1 = f(\alpha_1)$;
(4) Let $\tilde{\delta} = (\mathrm{denom}(\nu_0), \mathrm{denom}(\nu_1))$;
(5) Let $\tilde{\kappa} = \gcd(\tilde{\delta}\nu_0, \tilde{\delta}\nu_1)$;
(6) Return $\omega = \tilde{\delta}/\tilde{\kappa}$
```

Theorem 11. *With probability at least* $1/2$ *the output* ω *of* `DenomAndCont` (f, μ) *is such that the content of* ωf *is* μ*-smooth.*

Proof. In step (2) we simply find a small non-zero evaluation point for f. We expect that at most two evaluations of f are required.

In step (3) we approximate the denominator δ of f. Suppose $\overline{f} = \delta f$. For any prime $p \mid \delta$ we know that $\overline{f} \not\equiv (0) \bmod p$ (since δ is relatively prime to the content κ of f). For any prime p, the number of $\alpha_1 \in \{0, \ldots, p-1\}$ for which $\overline{f}(\alpha_1) \equiv 0 \bmod p$ is at most D. For $p \geq \mu$, the number of $\alpha_1 \in \{0, \ldots, \mu^2 - 1\}$ such that $f(\alpha_1) \equiv 0 \bmod p$ is at most $D\mu$. The number of prime divisors of δ is less than $\log H$. Thus, with probability at most $D\log H/\mu$ we choose an α_1 such that $\overline{f}(\alpha_1) \equiv 0 \bmod p$ for some $p \geq \mu$ which divides δ. By our choice of μ this probability is less than $1/4$.

In step (5) we approximate the content κ of δf. Suppose that $\tilde{\delta} f$ has content $\tilde{\kappa}$. We know $\tilde{\kappa}$ is κ times some μ-smooth number, and $\tilde{\delta} f = \tilde{\delta}\tilde{\kappa} f_0$, where f_0 is primitive. Clearly $\kappa \mid \nu_1$. For any prime p, the number of $\alpha_1 \in \mathcal{L}$ for which $f_0(\alpha_1) \equiv 0 \bmod p$ is at most D. For any prime $p \geq \mu$, the number of $\alpha_1 \in \{0, \ldots, \mu^2 - 1\}$ such that $f_0(\alpha_1) \equiv 0 \bmod p$ is at most $D\mu$. Now $\mathrm{numer}(\nu_0) < \mu^{2D+2}H$, and has at most $\log_\mu(\mu^{2D+2}H) < \log H + 2D + 2$ prime factors $p \geq \mu$. Thus, the probability that we choose an α_1 such that there is any prime $p \geq \mu$ dividing $f_0(\alpha_1) \equiv 0 \bmod p$ is at most $D(\log H + 2D + 2)/\mu$. By our choice of μ this probability is less than $1/4$.

Thus, the overall probability of success is at least $1/2$ on any iteration. □

Once we have the $\omega =$ `DenomAndCont`(f), it is easy to construct a black box for the μ-primitive part by multiplying the result of an evaluation of f by ω.

Finding sparsest shifts of integer polynomials

Suppose we have a black box for a rational polynomial $f \in \mathbb{Q}[x]$, and a bound $D \geq d = \deg f$. We now describe the complete algorithm for finding a sparsest shift of f.

We first approximate the content to within a μ-smooth multiple using `DenomAndCont` (μ will be specified later). We then build a new black box for the μ-primitive part of f (by dividing out the content and denominator) and so assume from now on that f is μ-primitive.

As discussed earlier, when we run the Berlekamp/Massey algorithm on the sequence of polynomials $\{f(y^i - z)\}_{i\geq 1}$, we are really just constructing the discrepancy polynomials $\Delta_i(z, y)$ for $i = 1, 2, \ldots, t$. When we choose a random p and s and run Berlekamp/Massey on $\{f(p^i - s)\}_{i\geq 1}$ we are evaluating the discrepancy polynomials at (s, p). That is, the Berlekamp/Massey algorithm gives us a black box for the discrepancy polynomials. `FindLinFac` will be just what we need to find the smallest t such that $\Delta_{2t-1}(z, y)$ has a factor in z alone (at least in the case when $t \leq (d + 1)/2$).

We now examine the discrepancy polynomials more closely, and for $1 \leq i \leq t$ let $\alpha_i(z, y) = f(y^i - z)$ and

$$\overline{A}_i = \begin{pmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_i \\ \alpha_2 & \alpha_3 & \iddots & \alpha_{i+1} \\ \vdots & \iddots & \iddots & \vdots \\ \alpha_i & \cdots & \cdots & \alpha_{2i-1} \end{pmatrix} \in \mathbb{Q}[z, y]^{i \times i}, \qquad \nabla_i = \det \overline{A}_i.$$

The $(2i - 1)$st discrepancy of the sequence $\{\alpha_i\}_{i \geq 1}$ is $\Delta_{2i-1} = \nabla_i / \nabla_{i-1}$ for $i \geq 1$ (taking $\nabla_0 = 1$). The sparsest shift of f occurs when there exists an $s \in \mathbb{Q}$ (or perhaps an algebraic extension of $\mathbb{Q}$) such that $\Delta_{2t-1}(y, s) = 0$, i.e. when Δ_{2t-1} has a factor in z alone.

When $t \leq (d + 1)/2$, the sparsest shift is rational and unique, so we can apply the algorithm `FindLinFac` to the numerators in the Berlekamp/Massey algorithm to find the sparsest shift.

Theorem 12. *Given a black box for a μ-primitive polynomial $f \in \mathbb{Q}[x]$ of degree d, which we assume has a t-sparse shift $s \in \mathbb{Q}$, where $t \leq (d + 1)/2$, we can find $s \in \mathbb{Q}$ with an expected* $10t$ *evaluations of the black box.*

Proof. It is straightforward to show the bounds

$$\begin{aligned} &\|\nabla_i\| \leq i^i \cdot 2^{id}(1 + d)^i(1 + di)^i \cdot \|f\|^d, \\ &|b| \leq 2^t \cdot t^{2t} \cdot \|f\|^t, \\ &|a| \leq 2^t \cdot t^{2t} \cdot \|f\|^t \cdot (2d)^{dt}. \end{aligned}$$

Now use the algorithm `FindLinFac` on each discrepancy in turn. By Theorem 7, at the $(2t + 1)$st discrepancy we will find a, b such that $az - b$ divides $\Delta_t(z, y)$ with probability $1/5$ on any invocation. The sparsest shift is then $a/ - b$. By running the algorithm repeatedly, we expect to find t and s with $5t$ invocations of `FindLinFac`, i.e. using $10t$ sequences. □

The cost of the algorithm is again dominated by the Berlekamp/Massey algorithm on the sequences $f(\gamma_1^i - \sigma)$ and $f(\gamma_2^i - \sigma)$ for $i = 0, \ldots$. The rational numbers involved do not have more than $O(d\tau^2 \log\|f\|)$ bits, where $d = \deg f$. Thus, the total cost is bounded by $O(\tau^2 \mathsf{M}(d\tau^2 \log\|f\|))$ bit operations, where now $O(\mathsf{M}(\ell))$ bit operations are sufficient to multiply two integers with ℓ bits. Again, as in the polynomial case, $\mathsf{M}(\ell) = \ell^2$ using the standard algorithm, and $\mathsf{M}(\ell) = \ell \log \ell \log\log \ell$ using asymptotically fast arithmetic.

All the notes following Theorem 7 apply here. In fact we heuristically expect that only one invocation of the algorithm will be needed to achieve success.

Once we find a sparsest shift, the polynomial can be recovered by completing the Ben-Or/Tiwari algorithm steps with the evaluations and generator already computed. Therefore, we regard this algorithm as an improved sparse interpolation algorithm: it *discovers and interpolates* with respect to a possible sparsest basis during the interpolation procedure.

The “one projection, one sequence” algorithm for univariate polynomials of Section 3.2 holds even more promise when a second “shift” projection is used. That is, we proceed as in `FindLinFac`, but instead of taking the GCD of the discrepancies of two different sequences, we take the GCD’s of the $(i-1)$st and ith discrepancies. As noted in Section 3.2, we conjecture this reveals the linear factor symbolically, and if this is indeed the case, we might hope that only *one* randomly shifted integer sequence is needed.

Multivariate rational polynomials with very sparse shifts

In the case when a polynomial $f \in \mathbb{Q}[x_1, \dots, x_n]$ has a unique “very sparse” shift, we can in fact reduce the problem of computing this sparse shift to the univariate case. In particular, by a “very sparse” shift, we mean one such that meets the criteria of Lemma 3: the minimal sparsity τ after this shift is at most $(\delta + 1)/2$, where $\delta = \min_{1 \le i \le n} d_i$ and $d_i = \deg_{x_i} f$ for $1 \le i \le n$.

In fact, we can make a stronger statement. Considering f as a polynomial in $\mathbb{Q}(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)[x_i]$, we define the sparsest shift of f in x_i as the $\theta_i \in \overline{\mathbb{Q}}$ such that when written in the power basis of $(x_i - \theta_i)$, f has the smallest number of non-zero coefficients (in $\mathbb{Q}[x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n]$). Define τ_i to be this minimal number of non-zero coefficients. Clearly $\tau_i \le \tau$ for all i $(1 \le i \le n)$.

Theorem 13. *Let $f \in \mathbb{Q}[x_1, \dots, x_n]$ have sparsest shift $(\theta_1, \dots, \theta_n) \in \mathbb{Q}^n$. As well, assume that for $1 \le i \le n$, the sparsest shift of f in x_i has sparsity $\tau_i \le (d_i + 1)/2$. For any i, $1 \le i \le n$, let $\mathcal{L}_i = \{0, \dots, 2d_i - 1\}$ and randomly choose $a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n \in \mathcal{L}_i$. The sparsest shift of $f(a_1, \dots, a_{i-1}, x_i, a_{i+1}, \dots, a_n)$ in x_i equals $\theta_i \in \mathbb{Q}$ with probability greater than $1/2$. It is not the sparsest shift only if $\deg_{x_i} f(a_1, \dots, a_{i-1}, x_i, a_{i+1}, \dots, a_n) < d_i$.*

Proof. Write f as

$$f = \sum_{0 \le j \le d_i} f_j^{(i)}(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) x_i^j.$$

The leading coefficient of $f(a_1, \dots, a_{i-1}, x_i, a_{i+1}, \dots, a_n)$ is $f_{d_i}^{(i)}(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n)$. This is non-zero with probability at least $1/2$ by the Schwartz–Zippel lemma. If this is indeed the case, $f(a_1, \dots, a_{i-1}, x_i, a_{i+1}, \dots, a_n)$ has a unique shift of sparsity less than $(d_i + 1)/2$. This shift must be θ_i. □

We can use this theorem to solve for the sparsest shift of a multivariate $f \in \mathbb{Q}[x_1, \dots, x_n]$ whenever the conditions are met. Simply find the sparsest shift in each variable in turn, using, for example, the two-projection algorithm described in the previous section.

3.4. Finding sparsest shifts in non-standard bases

The algorithms for finding sparsest shifts in non-standard bases are derived analogously to the early termination properties of sparse interpolation algorithms (Kaltofen and Lee, 2003).

Sparsest shifts in the Pochhammer basis

A univariate polynomial $f(x)$ can be represented in the Pochhammer basis as

$$f(x) = \sum_{j=1}^{t} u_j x^{\overline{d_j}} \qquad \text{and} \qquad u_j \neq 0,$$

with $x^{\overline{n}} = x(x+1)\cdots(x+n-1)$ for any integer $n \geq 0$. A sparsest shift $\theta \in S$ in the Pochhammer basis is an element in S such that t is minimized as τ in

$$f(x) = \sum_{j=1}^{\tau} c_j (x+\theta)^{\overline{e_j}} \qquad \text{and} \qquad c_j \neq 0,$$

and a T-sparse shift s is such that $t \leq T$.

Let $f^{(k)}(x) = \sum_{j=1}^{t} d_j^k u_j x^{\overline{d_j}}$ and $\Delta(f(x)) = f(x+1) - f(x)$. By the recurrence $f^{(k+1)}(x) = x \cdot \Delta(f^{(k)}(x))$, $f^{(k)}(x)$ can be obtained directly from $f(x), \ldots, f(x+2k-1)$. The early termination sparse interpolation in the Pochhammer basis of x (Kaltofen and Lee, 2003) is based on the following fact: the fraction-free Berlekamp/Massey algorithm first encounters a zero discrepancy after processing exactly $2t+1$ elements from the sequence $\{f^{(k)}(x)\}_{k \geq 0}$ (note that k starts from 0 here). The Pochhammer exponents of $f(x) = \sum_{j=1}^{t} u_j x^{\overline{d_j}}$ are the roots of the minimal generating polynomial $\Lambda(\zeta)$, namely, $\Lambda(\zeta) = \prod_{j=1}^{t}(\zeta - d_j) = \lambda_t \zeta^t + \lambda_{t-1}\zeta^{t-1} + \cdots + \lambda_0$.

To find the sparsest shifts, we introduce the shift variable z and consider $y = x + z$. The recurrence for $f^{(k)}(x)$ in the Pochhammer basis of y becomes $f^{(k+1)}(y-z) = y \cdot \Delta(f^{(k)}(y-z))$ with $\Delta(f(y-z)) = f(y+1-z) - f(y-z)$. We carry out the fraction-free Berlekamp/Massey algorithm on the sequence $\{f^{(k)}(y-z)\}_{k \geq 0}$: the discrepancies Δ_k are polynomials in y whose coefficients are polynomials in z. The solutions for z such that Δ_k first becomes the zero polynomial in y are the sparsest shifts θ. These occur at $k = 2\tau$, where τ is the number of terms in f in a sparsest shifted Pochhammer basis. Similarly, the T-sparse shifts are solutions for z in $\Delta_{2T} = 0$.

The special “tricks” discussed earlier for the univariate power bases can be implemented correspondingly. Yet, when applying the projection algorithms (see Sections 3.2 and 3.3), we need to assure $y = x + z$ is projected to a positive value. Moreover, consider a Pochhammer term projected to two different values p and q: $c_j \cdot p \cdot (p+1) \cdots (p+e_j-1)$ and $c_j \cdot q \cdot (q+1) \cdots (q+e_j-1)$. Due to the factorial nature, if $|p-q| < \deg f$, our projection algorithms might falsely include some of $1, \ldots, e_j - 1$ as shifts.

Sparsest shifts in the Chebyshev basis

Let $T_i(x)$ denote the ith Chebyshev polynomial of the first kind: $T_0(x) = 1$, $T_1(x) = x$, and $T_i(x) = 2xT_{i-1}(x) - T_{i-2}(x)$ for $i \geq 2$. A polynomial $f(x)$ can be represented in the Chebyshev basis:

$$f(x) = \sum_{j=1}^{t} u_j T_{d_j}(x) \qquad \text{and} \qquad u_j \neq 0.$$

A sparsest shift $\theta \in S$ is an element in S such that t is minimized to τ in

$$f(x) = \sum_{j=1}^{\tau} c_j T_{e_j}(x+\theta) \qquad \text{and} \qquad c_j \neq 0.$$

The early termination sparse interpolation in the Chebyshev basis of x (Kaltofen and Lee, 2003) introduces a symbolic p_c and interpolates $\tilde{f} = f + p_c$ which has exactly $\tilde{t}$ non-zero terms (f is recovered by removing p_c from $\tilde{f}$ at the end). If $\tilde{\alpha}_k(x) = \tilde{f}(T_k(x))$, the matrix

$$\tilde{\mathcal{A}}_k(x) = \begin{bmatrix} 2\tilde{\alpha}_0 & 2\tilde{\alpha}_1 & \dots & 2\tilde{\alpha}_{k-1} \\ 2\tilde{\alpha}_1 & \tilde{\alpha}_2 + \tilde{\alpha}_0 & \dots & \tilde{\alpha}_k + \tilde{\alpha}_{k-2} \\ \vdots & \vdots & \ddots & \vdots \\ 2\tilde{\alpha}_{k-1} & \tilde{\alpha}_k + \tilde{\alpha}_{k-2} & \dots & \tilde{\alpha}_{2k-2} + \tilde{\alpha}_0 \end{bmatrix} \tag{9}$$

is non-singular for $1 \leq k \leq \tilde{t}$, and singular for $k \geq \tilde{t} + 1$.

To find the sparsest shifts, we introduce the shift variable z and $y = x + z$. Unlike interpolation, we consider $\overline{f}(y-z) = f(y-z) + p_c T_{d_t}(y)$ with $p_c \neq -u_t$ such that f and $\overline{f}$ have exactly the same Chebyshev terms in y. Note that $d_t = e_\tau = \deg f = \deg \overline{f}$, $u_t = c_\tau$, and that f can be recovered by removing the added term whenever it is necessary. When $\overline{\alpha}_i = \overline{f}(T_i(y) - z)$ and the (i,j)th entry of $\overline{\mathcal{A}}_k(y,z)$ is $\overline{\alpha}_{i+j-2} + \overline{\alpha}_{|i-j|}$, the introduction of $p_c T_{d_t}(y)$ provides $\overline{\mathcal{A}}_k(y,z)$ being non-singular before k reaches the sparsity of $\overline{f}$ in the Chebyshev basis of y. That is, $1 \leq k \leq \tilde{t}$ (Kaltofen and Lee, 2003, cf. Theorem 11).

The sparsest shifts are the solutions for z such that the $\overline{\mathcal{A}}_k(y,z)$ first becomes singular, that is, the first k such that $\det \overline{\mathcal{A}}_k(y,z)$ is the zero polynomial in y. The singularities can also be detected at a zero discrepancy in the modified Gohberg/Koltracht algorithm (Kaltofen and Lee, 2003), so we can also choose to solve z such that the discrepancy first becomes the zero polynomial in y.

Finding the T-sparse shifts can be formulated similarly. However, one needs to take into consideration that the non-singularities of all principal leading submatrices are assumed in the modified Gohberg/Koltracht algorithm.

Treating p_c as a value, our additional "tricks" for the univariate power bases can be applied accordingly. Also, $y = x + z$ has to be projected to a value larger than one when applying the projection algorithms.

4. Extensions and improvements

4.1. *Prune the highest degree terms*

In addition to imposing a lower bound or an upper bound, when available, to the sparsities (see Section 2.2), we can also reduce the computations by pruning the highest degree terms.

Consider a univariate polynomial $f(x)$ in any two power bases:

$$\begin{aligned} f(x) &= u_1 x^{d_1} + u_2 x^{d_2} + \cdots + u_t x^{d_t} \\ &= c_1 (x+s)^{e_1} + c_2 (x+s)^{e_2} + \cdots + c_\tau (x+s)^{e_\tau}, \end{aligned}$$

with $u_i \neq 0$ for $1 \leq i \leq t$, $c_j \neq 0$ for $1 \leq j \leq \tau$, and $d_1 < d_2 < \cdots < d_t = \deg f$, $e_1 < e_2 < \cdots < e_\tau = \deg f$. The highest degree term remains unchanged, that is, its degree and coefficient are fixed in all shifted bases: $u_t = c_\tau$, $d_t = e_\tau = \deg f$.

In fact, for a multivariate polynomial $f(x_1, \ldots, x_n) = \sum_{i=1}^{t} u_i x_1^{d_{i,1}} \cdots x_n^{d_{i,n}}$, its highest degree terms in every direction will stay fixed in all shifted power bases; they are the terms with exponents $(d_{i,1}, \ldots, d_{i,n})$ such that for every $j \neq i$ there is a $\nu_k > 0$, $1 \leq k \leq n$, in $(\nu_1, \ldots, \nu_n) = (d_{i,1} - d_{j,1}, \ldots, d_{i,n} - d_{j,n})$.

For a given polynomial, if some or all highest degree terms are known, e.g. if the polynomial is known to be monic, let $\sum_{i=1}^{\kappa} c_i (y_1 - z_1)^{e_{i,1}} \cdots (y_n - z_n)^{e_{i,n}}$ sum up those terms. Now instead of $f(y_1 - z_1, \ldots, y_n - z_n)$, we proceed in our algorithms with

$$\overline{f} = f(y_1 - z_1, \ldots, y_n - z_n) - \sum_{i=1}^{\kappa} c_i (y_1 - z_1)^{e_{i,1}} \cdots (y_n - z_n)^{e_{i,n}}, \tag{10}$$

which has κ fewer terms than f in every shifted basis. Our algorithms for finding the sparsest shifts are all sensitive to the optimal sparsity. That is, instead of using the $(2\tau + 1)$st discrepancy $\Delta_{2\tau+1}$, the sparsest shifts can be recovered from the $(2\tau - 2\kappa + 1)$st discrepancy $\Delta_{2(\tau-\kappa)+1}$.

The "non-trivial GCD" trick in the univariate case can be further exploited: suppose the highest degree term in f, $c_\tau x^{e_\tau}$, is known. We can proceed with our algorithm with $f(y - z)$ and $\overline{f} = f(y - z) - c_\tau (y - z)^{e_\tau}$ to update their discrepancies Δ_i and $\overline{\Delta}_j$ accordingly. Since $\overline{f}$ has one term less than f, the sparsest shifts for f are the solutions to $\Delta_{2\tau+1} = \overline{\Delta}_{2\tau-1} = 0$, which can be computed through finding the first $\gcd(\Delta_{2i+1}, \overline{\Delta}_{2i-1})$ that is non-trivial in z. Note that in the multivariate case, we have a system of polynomial equations and each equation is a zero discrepancy corresponding to the polynomial pruned with a subset of its highest degree terms.

For the problem of finding T-sparse shifts, we can proceed with $\overline{f}$ in (10) and consider $\Delta_{2(T-\kappa)+1} = 0$ similarly. We note that the highest term pruning techniques can be applied to the Pochhammer and Chebyshev bases as well.

4.2. Finding sparsest shifts for a set of polynomials

Multivariate shifts within a designated set enable us to compute sparsest shifts that simultaneously minimize the terms of a given set of polynomials.

Consider a set of m polynomials $f_k(x_1, \ldots, x_n) \in \mathsf{D}[x_1, \ldots, x_n]$ for $1 \leq k \leq m$. An $s = (s_1, \ldots, s_n)$-shifted power basis represents f_k, $1 \leq k \leq m$, as

$$f_k(x_1, \ldots, x_n) = \sum_{j=1}^{t_k} c_{k,j} (x_1 + s_1)^{e_{k,1,j}} \cdots (x_n + s_n)^{e_{k,n,j}} = \sum_{j=1}^{t_k} c_{k,j} \beta_{k,j}^{e_{k,j}} \tag{11}$$

with $c_{k,i} \neq 0$.

There are a number of different ways to measure the sparsity of a set of polynomials.

Minimize the sum of the number of shifted terms

We look for all shifts $s \in S$ such that $t_1 + \cdots + t_k$ in (11) is minimized. Introduce $m - 1$ indeterminates and construct a polynomial F:

$$F(x_1, \ldots, x_n, \mu_1, \ldots, \mu_{m-1}) = \mu_1 f_1 + \mu_2 f_2 + \cdots + \mu_{m-1} f_{m-1} + f_m. \qquad (12)$$

Consider the shifts for F within $\overline{S} = (s_1, \ldots, s_n, 0, \ldots, 0)$ with $(s_1, \ldots, s_n) \in S$. The shifts $\theta = (\theta_1, \ldots, \theta_n)$ that minimize $t_1 + \cdots + t_n$ can be obtained by finding the sparsest shifts $\overline{\theta} = (\theta_1, \ldots, \theta_n, 0, \ldots, 0)$ for F within $\overline{S}$.

Although there are $m-1$ variables introduced, the shifts in those variables are fixed as 0. As a result, when using random projections for removal of variables in the discrepancies, all μ_k can be evaluated to scalars. An alternative is to use a single indeterminate μ_0 and find the sparsest shifts within $\overline{S}_0 = (s_1, \ldots, s_n, 0)$, where $(s_1, \ldots, s_n) \in S$, for the polynomial G:

$$G(x_1, \ldots, x_n, \mu_0) = \mu_0 f_1 + \mu_0^2 f_2 + \cdots + \mu_0^{m-1} f_{m-1} + f_m. \qquad (13)$$

We note that randomly projecting μ_0 may lead to larger scalars than before.

Minimize the number of distinct shifted terms

Here we want to minimize the number of distinct $\beta_{k,i}$ in (11) to represent every f_k, $1 \le k \le m$.

The polynomial F in (12) is now a polynomial in $x_1, \ldots, x_n$ over a coefficient domain $\mathsf{D}[\mu_1, \ldots, \mu_{m-1}]$. Supposing $k \neq l$, we assume $k < l \le m$ and consider $c_{k,i}\beta_{k,i}^{e_{k,i}}$ from f_k and $c_{l,j}\beta_{l,j}^{e_{l,j}}$ from f_l. If $\beta_{k,i}^{e_{k,i}} = \beta_{l,j}^{e_{l,j}}$, then their corresponding terms in $F(x_1, \ldots, x_n)$ collide into one term with coefficient either $\mu_k c_{k,i} + \mu_l c_{l,j}$ (when $l < m$) or $\mu_k c_{k,i} + c_{l,i}$ (when $l = m$), which cannot be a zero polynomial in μ_k. The sparsest shifts for $F \in \mathsf{D}[\mu_1, \ldots, \mu_{m-1}][x_1, \ldots, x_m]$ in the power basis of $x_1, \ldots, x_n$ thus minimize the number of distinct $\beta_{k,i}$ in representing $f_1, \ldots, f_m$. Similarly, we can work with a single indeterminate and compute the sparsest shifts for G in (13) in the power basis of $x_1, \ldots, x_n$ over $\mathsf{D}[\mu_0]$.

This method can be extended to a set of polynomials in the Pochhammer and Chebyshev bases.

Minimize the maximum of the number of shifted terms

Here we look for all shifts s in (11) such that $\max(t_1, \ldots, t_m)$ is minimized.

When performing the fraction-free Berlekamp/Massey algorithm on $\{f_k(x_1^i - s_1, \ldots, x_n^i - s_n)\}_{i \ge 0}$ for a polynomial f_k, the discrepancies $\Delta_{k,i}$ become zero for all $i \ge 2t_k + 1$. Now update $\Delta_{k,i}$ for each i in parallel until a solution $\theta = (\theta_1, \ldots, \theta_n)$ to the system $\Delta_{1,i}(\theta) = \cdots = \Delta_{m,i}(\theta) = 0$ is found.

This method simply performs the shift-finding algorithm for each polynomial in parallel, and can be applied to a set of polynomials in the Pochhammer and Chebyshev bases.

Acknowledgements

This material is based on work supported in part by Natural Sciences and Engineering Research Council of Canada, the Ontario Research & Development Challenge Fund, and Maplesoft (Giesbrecht, Lee) and by the National Science Foundation under Grant Nos. DMS-9977392, CCR-9988177 and CCR-0113121 (Kaltofen).

References

Ben-Or, M., Tiwari, P., 1988. A deterministic algorithm for sparse multivariate polynomial interpolation. In: Proc. Twentieth Annual ACM Symp. Theory Comput. ACM Press, New York, NY, pp. 301–309.

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

DeMillo, R.A., Lipton, R.J., 1978. A probabilistic remark on algebraic program testing. Inform. Process. Lett. 7 (4), 193–195.

Díaz, A., Kaltofen, E., 1998. FoxBox a system for manipulating symbolic objects in black box representation. In: Gloor, O. (Ed.), ISSAC'98, Proc. 1998 Internat. Symp. Symbolic Algebraic Comput. ACM Press, New York, NY, pp. 30–37.

Evans, R.J., Isaacs, I.M., 1976. Generalized Vandermonde determinants and roots of unity of prime order. Proc. Amer. Math. Soc. 58, 51–54.

Giesbrecht, M., Kaltofen, E., Lee, W.-s., 2002. Algorithms for computing the sparsest shifts for polynomials via the Berlekamp/Massey algorithm. In: Mora, T. (Ed.), ISSAC'02, Proc. 2002 Internat. Symp. Symbolic Algebraic Comput. ACM Press, New York, NY, pp. 101–108.

Grigoriev, D.Y., Karpinski, M., 1993. A zero-test and an interpolation algorithm for the shifted sparse polynomials. In: Proc. AAECC-10. Lecture Notes in Computer Science, vol. 673. Springer Verlag, Heidelberg, Germany, pp. 162–169.

Grigoriev, D.Y., Karpinski, M., Singer, M.F., 1990. Fast parallel algorithms for sparse multivariate polynomial interpolation over finite fields. SIAM J. Comput. 19 (6), 1059–1063.

Grigoriev, D.Y., Karpinski, M., Singer, M.F., 1994. Computational complexity of sparse rational function interpolation. SIAM J. Comput. 23, 1–11.

Grigoriev, D.Y., Lakshman, Y.N., 1995. Algorithms for computing sparse shifts for multivariate polynomials. In: Levelt, A.H.M. (Ed.), ISSAC'95, Proc. 1995 Internat. Symp. Symbolic Algebraic Comput. ACM Press, New York, NY, pp. 96–103.

Grigoriev, D.Y., Lakshman, Y.N., 2000. Algorithms for computing sparse shifts for multivariate polynomials. Appl. Algebra Engrg. Comm. Comput. 11 (1), 43–67.

Kaltofen, E., Lee, W.-s., 2003. Early termination in sparseinterpolation algorithms. J. Symbolic Comput., 40. In the special issues on papers of the 2003 Internat. Symp. Symbolic Algebraic Comput (2003) (doi: 10.1016/S0747-7171(03)00088-9).

Kaltofen, E., Lee, W.-s., Lobo, A.A., 2000. Early termination in Ben-Or/Tiwari sparse interpolation and a hybrid of Zippel's algorithm. In: Traverso, C. (Ed.), ISSAC'00, Proc. 2000 Internat. Symp. Symbolic Algebraic Comput. ACM Press, New York, NY, pp. 192–201.

Kaltofen, E., Trager, B., 1990. Computing with polynomials given by black boxes for their evaluations: greatest common divisors, factorization, separation of numerators and denominators. J. Symbolic Comput. 9 (3), 301–320.

Knuth, D.E., 1983. Mathematics for the Analysis of Algorithms, third ed. Birkhäuser.

Knuth, D.E., 1997. Seminumerical algorithms. In: The Art of Computer Programming, vol. 2, third ed. Addison Wesley, Reading, MA, USA.

Lakshman, Y.N., Saunders, B.D., 1996. Sparse shifts for univariate polynomials. Appl. Algebra Engrg. Comm. Comput. 7 (5), 364

Rosser, J.B., Schoenfeld, L., 1962. Approximate formulas for some functions of prime numbers. Ill. J. Math. 6, 64–94.

Schwartz, J.T., 1980. Fast probabilistic algorithms for verification of polynomial identities. J. ACM 27, 701–717.

Zippel, R., 1979. Probabilistic algorithms for sparse polynomials. In: Proc. EUROSAM’79. Lecture Notes Computer Science, vol. 72. Springer Verlag, Heidelberg, Germany, pp. 216–226.