Sparse Multivariate Function Recovery from Values with Noise and Outlier Errors*

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

Zhengfeng Yang
Shanghai Key Laboratory of Trustworthy Computing
East China Normal University
Shanghai 200062, China
zfyang@sei.ecnu.edu.cn

ABSTRACT

Error-correcting decoding is generalized to multivariate sparse rational function recovery from evaluations that can be numerically inaccurate and where several evaluations can have severe errors (“outliers”). The generalization of the Berlekamp-Welch decoder to exact Cauchy interpolation of univariate rational functions from values with faults is by Kaltofen and Pernet in 2012 [to be submitted]. We give a different univariate solution based on structured linear algebra that yields a stable decoder with floating point arithmetic. Our multivariate polynomial and rational function interpolation algorithm combines Zippel’s symbolic sparse polynomial interpolation technique [Ph.D. Thesis MIT 1979] with the numeric algorithm by Kaltofen, Yang, and Zhi [Proc. SNC 2007], and removes outliers (“cleans up data”) through techniques from error correcting codes. Our multivariate algorithm can build a sparse model from a number of evaluations that is linear in the sparsity of the model.

Categories and Subject Descriptors: I.1.2 [Symbolic and Algebraic Manipulation]; Algorithms; G.1.1 [Numerical Analysis]; Interpolation—smoothing

Keywords: error correcting coding, fault tolerance, Cauchy interpolation, rational function

1. INTRODUCTION

Reed-Solomon error correcting coding uses evaluation of a polynomial as the encoding device, and interpolation as the decoding device. The polynomial is oversampled, and up to $E$ errors in the evaluations are corrected via an additional $2E$ sample points. Blahut’s decoding algorithm [1], for evaluations at consecutive powers of roots of unity, locates the erroneous evaluations by sparse interpolation. Berlekamp/Welch decoding, for any set of distinct input arguments, reconstructs the error-corrected polynomial via Bezout coefficients in a polynomial extended Euclidean algorithm [23]. In [4] we address the situation when the polynomial is sparse: our decoding algorithm requires $2T(2E+1)$ evaluations for a polynomial with $t$ non-zero terms, when bounds $T \geq t$ and $E \geq k$ are input. Here $k$ is again the number of faulty evaluations, whose locations are unknown. We use the Prony/Blahut sparse interpolation algorithm and correct $k$ errors in a linearly generated sequence of evaluations; thus we perform $T^{1+o(1)}E$ arithmetic operations. Our algorithm is deployed on numerical data [4, Section 6] via the floating-point versions of the Prony/Blahut algorithm [8, 9, 18].

In [14] we have generalized the Berlekamp/Welch procedure to reconstructing a rational number or a univariate rational function over a field from multiple residues or evaluations, under the assumption that some residues and values are faulty. Again, we use the extended Euclidean algorithm. Our algorithms are for exact arithmetic.

In [17] we generalize Zippel’s interpolation algorithm for sparse multivariate polynomials [24] to sparse multivariate rational functions (cf. [6, Section 4]). We present a worst case analysis for exact arithmetic [17, Section 4.1], which for rational functions is more difficult than for polynomials, and implement the algorithm for noisy data with floating point arithmetic. The algorithm is numerically stable because 1. univariate rational recovery is accomplished by a structured total least norm algorithm on the original data, not by the extended Euclidean algorithm on derived data, and 2. multivariate recovery is performed on sparse candidates which constitute well-constrained models for loosely fitting data. As it turned out in [14], Berlekamp/Welch decoding is Cauchy interpolation, and the Euclidean algorithm computes an unreduced rational function.

We combine the insights from [17] and [14] and obtain the following:

1. a numerical, noise tolerant Berlekamp/Welch-like univariate polynomial and rational function interpolation algorithm that can remove outlier errors: our algorithm recovers the full coefficient vectors and for a sparse polynomial $f$ with $t \leq T$ terms requires $\deg(f) + 2E + 1$ evaluations compared to the $2T(2E+1)$ evaluations of the algorithm in [4, Section 6]. However, we can work with evaluations at arbitrary input arguments and can recover rational functions. Note that in [2, Section 3] we have shown stability for a numerical version of Blahut’s error-correcting polynomial interpolation algorithm for $E = 1$.
2. an exact interpolation algorithm for sparse multivariate polynomials and rational functions à la Zippel which can correct errors in the evaluations: in Section 2 we give an analysis which allows for evaluations at poles of the polynomial.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Cancellation Notice: ISSAC’13, June 26–29, 2013, Boston, Massachusetts, USA.
Copyright 2013 ACM 978-1-4503-2059-7/13/06 ...$15.00.
rational function. Errors in the evaluations may indicate false poles, and evaluations may falsely produce a value at a pole.

3. a numerical, noise tolerant interpolation algorithm for sparse multivariate polynomials and rational functions that can remove outlier errors. In least squares fitting of known models (note that our sparse models are computed by our algorithm), outliers can be identified by their leverage scores derived from the pseudo-inverse (projection matrix) of the normal equations. Our approach is entirely different, even for polynomial models. We locate those outliers via numerical error-correcting decoding, using structured linear algebra algorithms. Our computer experiments in Section 4 demonstrate that our approach is very feasible.

Remark 1.1. Reed-Solomon decoding reconstructs the coefficients of a polynomial \( f \) by interpolation where some of the evaluations are faulty. In fact, for \( d \) coefficients, i.e., \( \deg(f) \leq d - 1 \) and \( k \) errors, one needs \( L = d + 2E \) evaluations, where \( E \geq k \) bounds the number of faults \( \alpha \)-priori. In our algorithms, we will as substeps perform several interpolations, where each is designed to tolerate a given number \( E \) of errors. Since we view the acquisition of evaluations as probing a black box for the function \( f \), the error rate of the black box can be related to \( E \): Suppose the black box for any \( L \geq L_{\text{min}} \) evaluations produces faulty values for no more than \( k \leq L/q \) inputs, where \( q > 2 \). Here \( 1/q \) is the error rate, and \( L_{\text{min}} \) is a minimum on the number of each batch of evaluations: obviously, one cannot suppose that for \( L = 1 \) evaluation one always gets a correct answer. Then \( E = \lfloor d/(q-2) \rfloor \) yields \( L/q = (d + 2E)/q \leq (d + 2d)/(q - 2)/q = d/(q - 2) \), so \( k \leq E \) as required. For our multivariate algorithm, the situation is somewhat different, and the error rate of the black box for \( f/g \) cannot be too high: see Remark 2.6.

Remark 1.2. Since our evaluations are numerically inaccurate, a question arises at what point a noisy value becomes an outlier. Outliers give rise to a common univariate polynomial factor, the error locator polynomial, in the sparse multivariate rational function reconstruction of the model (see the discussion after Assumption 5). If sufficiently large in magnitude, they markedly increase the numerical rank of the corresponding matrix \( (15) \). Their locations can be determined by their corresponding black box inputs being roots of the error locator polynomial factor \( (22) \), which must be present. See also Remark 3.1 below.

As in \([17]\), our multivariate algorithm takes advantage of multivariate sparsity, and a stable univariate algorithm for dense fractions, now with error correction. Cauchy interpolation \([14]\) recovers the reduced fraction \( (f/GCD(f,g))/gcd(g,GCD(f,g)) \). It was first observed in \([19]\) that an unreduced fraction, e.g., \( (x^d - y^d)/(x - y) \), can yield a much sparser model for the black box. Such models can be constructed by interpolation, both for exact and for numeric data. In \([19]\) we give an exact univariate algorithm. In Kaltofen's ISSAC 2011 presentation at the FCRC in San Jose the use of \([3]\) on numeric data was introduced. Example 4.1 shows the feasibility of that approach. Note that the number of evaluations in \([19]\) depends logarithmically on the degrees.

2. ERROR-CORRECTING MULTIVARIATE RATIONAL FUNCTION INTERPOLATION

Here we generalize the analysis for exact arithmetic in \([17, \text{Section 4.1}]\). Consider the rational function \( f/g \in K(x_1, \ldots, x_n) \), where the numerator and denominator are represented as

\[
f = \sum_{j=1}^{t_f} a_j \bar{x}^d_j, \quad g = \sum_{m=1}^{t_g} b_m \bar{z}^e_m, \quad a_j, b_m \in K \setminus \{0\},
\]

where \( K \) is an arbitrary field and the terms are denoted by \( \bar{x}^d = x_1^{d_1} \cdots x_n^{d_n} \) and \( \bar{z}^e = z_1^{e_1} \cdots z_n^{e_n} \). We analyze our variant of Zippel’s sparse interpolation technique to recover the numerator and denominator. Zippel’s technique \([12, \text{Section 4}]\) determines the support of \( f_1 = f(x_1, \ldots, x_1, \alpha_1+1, \ldots, \alpha_n) \) and \( g_1 = g(x_1, \ldots, x_1, \alpha_1+1, \ldots, \alpha_n) \) iteratively from the support of \( f_{-1} \) and \( g_{-1} \), where \( \alpha_2, \ldots, \alpha_n \in K \) is a random anchor point. We will use Zippel’s probabilistic assumption.

Assumption 1. Each term \( x_1^{d_1} \cdots x_l^{d_l} \), where \( 1 \leq j \leq t_f \), and each term \( x_1^{e_1} \cdots x_l^{e_l} \), where \( 1 \leq m \leq t_g \), has a non-zero coefficient in \( f_{-1} \) and \( g_{-1} \).

Note that for different \( j \)'s and different \( m \)'s one may have the same term prefix in \( i - 1 \) variables. At this point we do not assume that \( f \) and \( g \) are relatively prime, but we will introduce relative primeness as Assumption 5 below for decoding; see also Remark 2.7.

We wish to recover \( f_1 \) and \( g_1 \) from the sparse supports of \( f_{-1} \) and \( g_{-1} \) and evaluations of \( f(x_1, \ldots, x_1)/g(x_1, \ldots, x_1) \) for \( \alpha_1+1, \ldots, \alpha_n \). We chose \( \xi_1, \ldots, \xi_l \in K \) and evaluate at powers \( \xi_1^i, \ldots, \xi_l^i \), \( \ell \in \{0, 1, 2, \ldots\} \). We will obtain

\[
\beta_{i,\ell} = \gamma_{i,\ell} + \gamma'_{i,\ell}, \quad \text{where} \quad \gamma_{i,\ell} = \frac{f(\xi_1^i, \ldots, \xi_l^i)}{g(\xi_1^i, \ldots, \xi_l^i)} \in K \cup \{\infty\}
\]

and where \( \gamma'_{i,\ell} \neq 0 \) exactly at the \( k \leq E \) unknown indices \( 0 \leq \lambda_1 < \lambda_2 < \cdots < \lambda_k \) for \( \ell \), that is \( \gamma_{i,\ell} = 0 \) for all \( \ell \notin \{\lambda_1, \ldots, \lambda_k\} \).

Assumption 2. We assume that we have the upper bound, \( E \), on the number of erroneous evaluations, but not the actual count of errors, \( k \), and not their locations \( \lambda_k \).

If \( g(\xi_1^i, \ldots, \xi_l^i) \neq 0 \) we have \( \gamma_{i,\ell} = \infty \), but \( \beta_{i,\ell} \) can be erroneously \( \in K \). Similarly, if \( g(\xi_1^i, \ldots, \xi_l^i) = 0 \) we may erroneously have \( \beta_{i,\ell} = \infty \).

Following the Berlekamp/Welch strategy, we attempt to recover \( (f/\Lambda_i)/(g/\Lambda_i) \) where

\[
\Lambda_i = (x_1 - \xi_1^\lambda_1) \cdots (x_1 - \xi_1^\lambda_k)
\]

is an error locator polynomial. The set of possible terms in \( f/\Lambda_i \) and \( g/\Lambda_i \) can now be restricted to

\[
D_{f,\ell,E} = \{x_1^{d_1+\nu} \cdots x_l^{d_l+\nu} \mid 1 \leq j \leq t_f, 0 \leq \nu \leq E, \\
0 \leq \delta_j \leq \min(\deg(f) - d_{j,1} - \cdots - d_{j,\ell-1}, \deg_x(f))\}
\]

and

\[
D_{g,\ell,E} = \{x_1^{e_1+\nu} \cdots x_l^{e_l+\nu} \mid 1 \leq \nu \leq E, \\
0 \leq \eta_m \leq \min(\deg(g) - e_{m,1} - \cdots - e_{m,\ell-1}, \deg_y(g))\}
\]

Note again that not all of the terms enumerated in (4) and/or (5) are distinct. See Remark 2.4 below for somewhat smaller candidate term sets. Here we make the assumption that the \( f_{-1} \) and \( g_{-1} \), as said earlier, contain the full set of possible terms, that with high probability as we will inductively argue.

Assumption 3. We assume that we know \( \deg(f), \deg_x(f), \deg(g), \) and \( \deg_y(g) \).

Let \( \bar{g} \) and \( \bar{z} \) be the coefficient vectors of \( f/\Lambda_i \) and \( g/\Lambda_i \) for the (distinct) terms in \( D_{f,\ell,E} \) and \( D_{g,\ell,E} \). For any \( \ell = 0, 1, 2, \ldots \)
and any point \( \xi_1, \ldots, \xi_n \in K \) each value \( \beta_{i,t} \) in (2) constitutes a linear equation for the coefficient vector,
\[
\sum_{j,v,\delta} y_{j,v,\delta}(\xi_1^{d_{j,1}+v}, \ldots, \xi_n^{d_{j,n}+v}) = \beta_{i,t} \sum_{m,v,\eta} z_{m,v,\eta}(\xi_1^{e_{m,1}+v}, \ldots, \xi_n^{e_{m,n}+v}) \cdot \xi_i^t.
\]

Errors in the \( \beta_{i,\lambda} \) are tolerated because \( f, A_1 \) and \( g, A_4 \) are both \( = 0 \) at \( x_i = \xi_i^\kappa \). Note again that coefficients/indeterminates can be the same, \( y_{j,v,\delta} = y_{j2,v,\delta} \), for instance, if corresponding terms are the same.

With \( \ell = 0, \ldots, L - 1 \), where \( L \) is yet to be determined, the equations (6) form a homogeneous linear system in the unknowns \( y_{i,v,\delta} \) and \( z_{i,m,v,\eta} \),
\[ V_{i,L}(\xi_1, \ldots, \xi_n) \gamma^T = \Gamma_{i,L} \omega_i \omega_i^T, \]
where \( V_{i,L} \) is a diagonal matrix of rational function values \( \beta_{i}, \lambda, V_i \) and \( \omega_i \) are (transposed) Vandermonde matrices, with possible zero rows. If \( \beta_{i,\lambda} = \infty \) then the \( \ell \)-th row in \( V_{i,L} \) is set to a zero row, and \( (\Gamma_{i,L}, \ell) \) is set to 1.

Provided the term supports of \( f_{i-1} \) and \( g_{i-1} \) were correctly computed in the previous iterations, the coefficient vectors \( \{ \gamma_i, \cdots, \gamma_i \} \) of \( f_{i} \) and \( g_{i} \) solve (7). For the term sets in (4) and (5) let
\[ D_{f_1,E} = \{ \gamma_{1} \neq \gamma_{2} \neq \gamma_{3} \neq \gamma_{4} \neq \gamma_{5} \neq \gamma_{6} \}, \]
with \( |D_{f_1,E}| = |D_{g_1,E}| \). Now set \( L \geq |D_{f_1,E} \times D_{g_1,E}| \) in (7). We argue that for random \( \xi_1, \ldots, \xi_n \), the polynomials \( f \) and \( g \), which correspond to any non-zero solution vector \( \{ \gamma_i, \cdots, \gamma_i \} \), respectively, of the linear system (7), with high probability satisfy \( \gamma_{f_1} = \gamma_{g_1} \).

We shall first assume that the random choices for \( \xi_1, \ldots, \xi_n \in S \subset K \) are such that no two distinct terms in \( D_{f_1,E} \) and no two distinct terms in \( D_{g_1,E} \) evaluate at \( x_{\mu} = x_{i,\lambda} \), \( 1 \leq \mu \leq i, \) to the same element in \( K \). Because
\[ \forall \ell, 0 \leq \ell \leq L - 1: \{ (f_{i-1},g_{i-1})(\xi_1, \ldots, \xi_n) = \beta_{i,\ell} (g_{i-1})(\xi_1, \ldots, \xi_n), \}
\]
we have
\[ \forall \ell, 0 \leq \ell \leq L - 1: \{ (f_{i-1},g_{i-1})(\xi_1, \ldots, \xi_n) = (f_{i-1},g_{i-1})(\xi_1, \ldots, \xi_n), \}
\]
Note that if \( \beta_{i,\ell} = 0 \) then \( (f_{i-1},g_{i-1})(\xi_1, \ldots, \xi_n) = (f_{i-1},g_{i-1})(\xi_1, \ldots, \xi_n) = 0 \) and if \( \beta_{i,\ell} = \infty \) then \( (f_{i-1},g_{i-1})(\xi_1, \ldots, \xi_n) = (f_{i-1},g_{i-1})(\xi_1, \ldots, \xi_n) = 0 \).

The possibly occurring terms of the polynomial \( f_{i-1} = \gamma_{i} g_{i-1} \) are contained in \( D_{f_1,E} \cup \{ f, g \} \). Note that for \( i = 1 \) we have \( L = |D_{f_1,E} \cup \{ f, g \} | = \deg_x(f) + \deg_y(g) + 1 + 2E \); see Remark 2.5 below.

Assumption 4. Finally, we assume that the random choices for \( \xi_1, \ldots, \xi_n \in S \subset K \) are such that no two distinct terms in \( D_{f_1,E} \cup \{ f, g \} \) evaluate to the same value, which subsumes our earlier assumption on the distinctness of term evaluations.

For \( L \geq |D_{f_1,E} \cup \{ f, g \} | \) we then must have
\[ f_{i-1} \gamma_i = \gamma_i g_{i-1} = 0, \]
(10)
because the coefficient vector of \( f_{i-1} = \gamma_i g_{i-1} \) is by (9) a kernel vector in a square non-singular (transposed) Vandermonde matrix and therefore must be zero. Since \( f_{i-1} \neq 0 \) and \( g_{i-1} \neq 0 \), we have \( \gamma_i = 0 \). We have concluded the analysis of the error correction property of our linear system. Next, we discuss the recovery of a sparse rational interpolant for \( f_i, g_i \).

In [17] we have excluded 0 and \( \infty \) from the evaluations \( \gamma_i \) in (2), but here we show that those values are perfectly allowable. Our (generalized) Berlekamp/Welch decoding algorithm concludes as follows, at least for reduced fractions (1); see also Remark 2.7.

Assumption 5. GCD\( (f, g) = 1 \) in \( K[x_1, \ldots, x_n] \) in (1).

Suppose now the \( f_i, g_i \) are relatively prime in \( K[x_1, \ldots, x_i] \). For random anchor points \( \alpha_2, \ldots, \alpha_n \) this will be true with high probability. In fact, \( f_i, g_i \) are then random projections of the primitive parts of \( f, g \) after removing their contents in \( K[x_{i+1}, \ldots, x_n] \); see Remark 2.4. So we obtain from (10)
\[ f_i/g_i \neq \gamma_i, \]
(11)
by removing a common factor \( h \in K[x_1] \) of \( f \) and \( g \). Because of the degree constraints in the term supports in (4) and (5), no additional polynomial factors in the variables \( x_1, \ldots, x_i \), are possible. We thus have \( h f_i = f \) and \( h g_i = g \). All \( x_i - \xi_i^\kappa \) for all \( \kappa = 1, \ldots, k \) divide \( h(x_i) \), because by (11)
\[ \xi_i^\kappa \left\{ \xi_1^\kappa, \ldots, \xi_n^\kappa \right\} = \beta_{i,\kappa} g_i(\xi_1^\kappa, \ldots, \xi_n^\kappa), \]
but \( f_i(\xi_1^\kappa, \ldots, \xi_n^\kappa) \neq \beta_{i,\kappa} g_i(\xi_1^\kappa, \ldots, \xi_n^\kappa) \). Because we estimate the number of errors by \( E \) in (4) and (5), \( f \) and \( g \) can have common factors in \( K[x_1] \) in addition to \( h(x_i) \). Those common factors give the nullspace of (9) a dimension \( 1 + E \) to \( k \), and the kernel vectors corresponding to the lowest degree polynomials \( f \) and \( g \) in \( x_1 \) are the coefficient vectors of \( f = c f_i A_1 \) and \( g = c g_i A_1 \) for \( c \in K, c \neq 0 \).

Remark 2.1. Note that for \( g = 1 \) we obtain a sparse multivariate polynomial interpolation algorithm with error-correction, and Assumption 5 is satisfied. For the exact problem for multivariate polynomials, say with \( K \) a finite field, we also mention [22], where the minimum number of points is studied for unique recovery. There encoding/decoding is performed by combinatorial search for the erroneous evaluations.

\[ \hat{\gamma}_i = \gamma_i, \]
(12)
where \( \hat{\gamma}_i \) is the correct value of \( \gamma_i \).

Remark 2.2. The linear system (7) of linear constraints (6) may yield \( f_i \) and \( g_i \) for smaller \( L \). In [17] we have suggested to increment \( L \) until a 1-dimensional kernel is achieved. If \( k = E \) such a strategy would work here. The system (7) has \( M = |D_{f_1,E}| + |D_{g_1,E}| \) variables, so at least \( L \geq M - 1 \) equations are needed. In fact, from earlier iterations one has additional linear constraints for \( \mu = 1, 2, \ldots, i - 1 \):
\[ \sum_{m,v,\eta} z_{m,v,\eta}(\xi_1^{em,\nu}, \ldots, \xi_n^{em,\nu}) \alpha_{\mu+1} \cdots \alpha_i = \gamma_{\mu,\ell} \times \]
(12)
also could use fresh values $(\xi_1, \ldots, \xi_t)$. The proof of property (10) then uses the idea that for symbolic values $\xi_0 = x_i$, the property is true over the function field $K(x_1, \ldots, v_i)$. □

Remark 2.3. At iteration $i$ we have arbitrarily chosen the variable $x_1$ for our error-locator polynomial $\Lambda_i$. We could also have chosen $x_2$, or $x_3$, \ldots, or $x_t$. Again, one may select that variable $x_u$ for which the sets $D_{f,i,E}$ and $D_{g_E}$ have the fewest elements. Clearly, if $x_1$ occurs sparsely and $x_2$ densely, $x_2$ is likely a better choice. Note that if $x_1$ is chosen, one gets no overlap in the terms in $D_{f,i,E}$ or $D_{g_E}$. The variable-by-variable interpolation depends on a variable order. Different orders may lead to a different number of evaluations. For numerical reasons, one should process the variables with smaller degrees first; see Remark 2.7.

For the record, we give an explicit worst case estimate for the exact algorithm. If we denote by $t_{f,i} = |D_{f,i,0}|$ and $t_{g,i} = |D_{g,i,0}|$, the number of terms in $f_i$ and $g_i$ respectively, with $t_{f,0} = t_{g,0} = 1$, and if we chose the variable $x_i$ for $\Lambda_i$, one probably needs at most $(\log\delta_i - \sum_{j} \log(\delta_i) + \log(\delta_i) + 2E + 1) \leq \log(n - 1) + t_{f,i} + 1) \cdot \max(\log(\delta_i) + \log(\delta_i) + 2E + 1) \cdot \beta_i (x_i)$ (with high probability).

Remark 2.4. The term sets $D_{f,i,E}$ in (4) for $f_i\Lambda_i$ and $D_{g,i,E}$ in (5) for $g_i\Lambda_i$ should be as small as possible. One may restrict $\delta_i$ (4) and $\eta_m$ in (5) by $\delta_i \leq \deg(f_i) - d_i, \cdots, -d_{j,1}-1, \eta_m \leq \deg(g_i) - e_m, \cdots, -e_{m-1}$, where $\log(\delta_i) \approx \log(\delta_i)$ and $\log(\delta_i)$ is not sufficiently small. We have $|D_{f,i,E}| \leq (E + 1)|D_{f,i,0}|$ and $|D_{g,i,E}| \leq (E + 1)|D_{g,i,0}|$ in (4) and (5), so in practice the worst-case $L(E) \leq (E + 1)|D_{f,i,0}| + (E + 1)|D_{g,i,0}| < \log q \Rightarrow q > |D_{f,i,0}| + |D_{g,i,0}|$. □

Remark 2.5. Our initialization for $i = 1$ uses for $f_i$ all terms $x_1^\ell$, where $\delta = 0, 1, \ldots, \deg_a(f_i)$, and for $g_i$ all terms $x_1^n$, where $\eta = 0, 1, \ldots, \deg_a(g_i)$. By our definitions (10), (4), and (8) we evaluate the fraction at $\xi_1^\ell$ for $\ell = 0, 1, \ldots, L - 1$ with $L = \deg_a(f_i) + \deg_a(g_i) + 1 + 2E$. We suppose that $\xi_1^\ell \neq \xi_1^\ell$ in the range for $\ell$. Our algorithm in the initialization phase essentially implements Berlekamp/Welch decoding at Blahut points for rational functions, and in the above we have proved that the errors are removed, that without appealing to the Euclidean algorithm. The linear system approach for Berlekamp/Welch decoding is also introduced in [21]. □

Remark 2.6. As in Remark 1.1 for univariate interpolation, we can determine $E$ from the error rate $1/q$ of the black box for $f$ and $g$. For $L(E) \leq |D_{f,i,0}| + |D_{g,i,0}|$, or, in practice, $L(E) \leq |D_{f,i,E}| + |D_{g,i,E}| + L_0$, we use which we establish in Sections 3 and 4 with $L_0 = 10$, we must attain $k \leq L(E)/q \leq E$. Note that the latter may for $i \geq 2$ not have a solution for $E$ if the rate $1/q$ is not sufficiently small. We have $|D_{f,i,E}| \leq (E + 1)|D_{f,i,0}|$ and $|D_{g,i,E}| \leq (E + 1)|D_{g,i,0}|$ in (4) and (5), so in practice the worst-case $L(E) \leq (E + 1)|D_{f,i,0}| + (E + 1)|D_{g,i,0}| + L_0 \leq q \Rightarrow q > |D_{f,i,0}| + |D_{g,i,0}|$. □

Remark 2.7. The first algorithm for recovering a sparse rational function without Assumption 5 is described in [19]. As an example, the unordered fraction $(x^d - y^d)/(x - y)$ is much sparser than the reduced polynomial. In fact, in [19] univariate fractions are recovered as sparse fractions, not using dense Cauchy interpolation; the number of evaluation points in the algorithms is proportional to $\log(\deg(f_i))$. Here we have followed the idea of lifting an unordered fraction by delaying Assumption 5 until after establishing the key Berlekamp/Welch property (10), which is $f_i/g_i = f/j/g$. If $f_i/g_i$ is unordered, the error corrected $\tilde{f}$ and $\tilde{g}$ may not be equal to the sparse projections $f_i$ and $g_i$. As we have supposed in [19], the sparsest possibly unordered reduction $f/g$ of lowest degree can be unique, hence liftable via $\tilde{f}$ and $\tilde{g}$. The initial sparse $f_i$ and $g_i$ can be also obtained by computing a sparse polynomial multiple [10]. Numerically, it may also be possible to recover a sparse unordered fraction for $i = 1$ by optimizing the $1$-norm of the solution vector via linear programming [3]. Example 4.1 below demonstrates such a recovery. Such sparse unordered recovery is also useful when the evaluations at $\alpha_0$ (see Remark 2.4) do not yield a numerically relatively prime univariate pair $f_i, g_i$.

In the exact case, there are other ways of determining the coefficients in $K[x_i]$ of $f_i$ and $g_i$, for example by interpolating or sparsely interpolating $x_i$, which yields a smaller linear system and possibly fewer evaluations (cf. [24]). One may also reconstruct the fraction using Strassen’s removal of divisions approach: see [5] (cf. [11, end of Section 7] and [13, Section 4]). [5] recovers the sparse homogeneous parts from highest to lowest degree. Since their algorithm and Algorithm Black Box Numerator and Denominator in [15] are based on univariate Cauchy interpolation, any black box error rate $1/q < 1/2$ can be handled by those methods.

[5] does not address the problem of projections leading to a reducible univariate fraction. Especially in the numeric setting, approximate relative primeness of the projections is difficult to maintain throughout each univariate Cauchy
Our sparse system (7) is set up to avoid the reducedness requirement all together. The sparsity constraints numerically stabilize the algorithm, provided one starts with a correct term support for \( f \) and \( g \). By using more than one random anchor \( \alpha_j \), where \( 2 \leq \mu \leq n \), one can improve the probability that no occurring term is falsely dropped from the term sets for \( f \) and \( g \). □

**Remark 2.8.** After recovering the \( K[x_j] \) coefficients of \( f \) and \( g \), one may sparsify those coefficients by shifting \( x_i = x_i + \sigma_i \), where \( \sigma_i \) is either in \( K \) or algebraic over \( K \). See [7] for computing such a sparsifying shift exactly, and [2] for an algorithm that tolerates numerical noise (and outlier). □

**Remark 2.9.** One may interpolate several sparse rational functions with a known common denominator (or numerator) simultaneously with fewer evaluations by the above method. An algorithm for the exact univariate dense recovery problem (without erroneous values) is in [20]. □

3. NUMERICAL INTERPOLATION WITH OUTLIER ERRORS

Based on the discussion in Section 2, we present a modified Zippel’s sparse interpolation approach to recover sparse rational function from values with noise and outlier errors. In the approximate case, \( \Theta \) is introduced to measure whether the evaluation is an outlier error, that is, we say the evaluation \( \beta \) at the point \((\zeta_1, \ldots, \zeta_n) \in \mathbb{C}^n\) is an outlier error, if \( \beta = \gamma + \gamma' \), where \( \gamma = f(x_1, \ldots, x_n)/g(x_1, \ldots, x_n) \in \mathbb{C} \cup \{ \infty \} \), and \( |\beta|/|\gamma| \geq \Theta \). Again false poles and non-poles are allowed; see explanation immediately after Assumption 2. Consider the rational function \( f/g \in \mathbb{C}(x_1, \ldots, x_n) \), where \( f/g \) are represented as (1). Suppose a black box for \( f/g \) with noise and outlier errors at a known error rate is given. The upper bound on the number of erroneous evaluations \( E \) can be determined from the error rate; see Remark 1.1. in this Section, we at first present a method to interpolate a univariate rational function, and then discuss how to recover \( f \) and \( g \) when \( f_{i-1} \) and \( g_{i-1} \) are already computed.

Let \( f^{[i]} = f(\alpha_1, \ldots, \alpha_{i-1}, x_i, \alpha_i, \ldots, \alpha_n) = \sum_{j=1}^{d_f} \psi_j^{[i]} \rho_{ij}, \quad g^{[i]} = g(\alpha_1, \ldots, \alpha_{i-1}, x_i, \alpha_i, \ldots, \alpha_n) = \sum_{m=1}^{d_g} \chi_m^{[i]} \nu_{im}, \) and assume that (with high probability) the sets \( D_f^{[i]} \) and \( D_g^{[i]} \) at the end of Remark 2.4 are the corresponding nonzero terms of \( f^{[i]} \) and \( g^{[i]} \) Here we have a-priori total degree bounds \( d_f \geq \deg(f) \) and \( d_g \geq \deg(g) \). Now let us show how to compute those term supports \( D_f^{[i]} \) and \( D_g^{[i]} \) of the univariate polynomials \( f^{[i]} \) and \( g^{[i]} \) with respect to the variable \( x_i \). Our discussion is for \( i = 1 \). Given a random root of unity \( \zeta \in \mathbb{C} \), we compute the evaluations with outlier errors, that is, for \( \ell = 0, 1, \ldots, d_f + d_g + 2E + 1 \) we compute

\[
\beta_\ell = \gamma_\ell + \gamma'_{\ell}, \quad \text{where} \quad \gamma_\ell = \frac{f(\zeta^{\ell}, \alpha_2, \ldots, \alpha_n)}{g(\zeta^{\ell}, \alpha_2, \ldots, \alpha_n)} \in \mathbb{C} \cup \{ \infty \},
\]

where \( \gamma'_\ell \) denotes noise or possibly an outlier error. We have the upper bound of the number of erroneous evaluations \( E \), which means that the number of \( \ell \), such that \(|\beta_\ell/\gamma_\ell| \geq \Theta \), is \( \leq E \). Having (13), we construct the following linear equations for \( \ell = 0, 1, \ldots, d_f + d_g + 2E \):

\[
\sum_{j=0}^{d_f + E} y_j \zeta^{\ell j} - \beta_\ell \sum_{m=0}^{d_g + 2E} z_m \zeta^{\ell m} = 0,
\]

The above equations form a linear system

\[
G [\tilde{y} \quad \tilde{z}]^T = [V_1, -\Gamma_1 W_1] [\tilde{y} \quad \tilde{z}]^T = 0.
\]

where \( \Gamma_1 = \text{diag}(\beta_0, \beta_1, \ldots, \beta_{d_f + d_g + 2E}) \), and where \( V_1, W_1 \) are Vandermonde matrices generated by the vectors \([1, \zeta, \ldots, \zeta^{d_f + d_g + 2E}]^T \) and \([1, \zeta, \ldots, \zeta^{d_f + d_g + 2E}]^T \). The numerical rank deficiency of \( G \), denoted by \( \rho \), can be computed by checking the number of small singular values of \( G \) or finding the largest gap among the singular values. Suppose

\[
s = \min(d_f - \deg(x), \tilde{d}_f - \deg(x)) \implies |\tilde{d_f} - \tilde{d}_f| \geq s + k.
\]

According to the discussion following Assumption 5 in Section 2, we know that \( \rho = 1 + E - k + s \). Having \( \rho \), the linear equations (14) are transformed into the following reduced linear equations by removing some unknown coefficients of higher degree in (14), namely, for \( \ell = 0, 1, \ldots, d_f + d_g + 2E \)

\[
\sum_{j=0}^{d_f + E - \rho + 1} y_j \zeta^{\ell j} - \beta_\ell \sum_{m=0}^{d_g + E - \rho + 1} z_m \zeta^{\ell m} = 0,
\]

whence the matrix form is

\[
\tilde{G} [\tilde{y} \quad \tilde{z}]^T = [\tilde{V}_1, -\tilde{\Gamma}_1 \tilde{W}_1] [\tilde{y} \quad \tilde{z}]^T = 0.
\]

Note that the numerical rank deficiency of \( \tilde{G} \) is 1, since \( d_f + d_g + 2E - k = d - s + k \). The coefficient vector \( \tilde{y}^T \) of \( f^{[1]} \tilde{\Lambda} \) and the coefficient vector of \( g^{[1]} \tilde{\Lambda} \) are achieved from the last singular vector of \( \tilde{G} \). Note that \( \tilde{\Lambda} \) should have the form \( \tilde{A}_\ell = (x_1 - \zeta_1^{\ell 1}) \cdots (x_1 - \zeta_1^{\ell n}) \). In that case, every root \( \zeta_1^{\ell k} \), \( 1 \leq k \leq \rho \), of \( \tilde{\Lambda} \) can be detected by checking for \( \ell = 0, 1, \ldots, d_f + d_g + 2E \) with a preset tolerance \( \epsilon_{\text{root}} \):

\[
\ell \in \{1, \ldots, \lambda_k\} \iff |(f^{[1]} \tilde{\Lambda}_1)(\zeta_1^{\ell 1}) + (g^{[1]} \tilde{\Lambda}_1)(\zeta_1^{\ell n})| \leq \epsilon_{\text{root}}.
\]

Having \( \tilde{\Lambda}_1 \), we obtain \( f^{[1]} \) by applying the approximate univariate polynomial division technique between \( f^{[1]} \tilde{\Lambda}_1 \) with \( \tilde{\Lambda}_1 \). Similarly, \( g^{[1]} \) can be obtained by approximate polynomial division. In the end, the actual supports \( D_f^{[1]} \) and \( D_g^{[1]} \) corresponding to \( f^{[1]} \) and \( g^{[1]} \) can be obtained by removing the terms whose coefficients are in absolute value \( \leq \epsilon_{\text{coeff}} \). Performing the above technique for each variable \( x_i, 2 \leq i \leq n \), one may obtain all the nonzero terms \( D_f^{[1]} \) and \( D_g^{[1]} \) of \( f^{[1]} \) and \( g^{[1]} \).

**Remark 3.1.** The preset tolerance measures \( \epsilon_{\text{root}} \) and \( \epsilon_{\text{coeff}} \) require that the singular solution vector \( \tilde{y} \tilde{z} \) is normalized. We normalize the Euclidean 2-norm to 1. Because we oversample by \( d_f - \deg(x), f + d_g - \deg(x) \) evaluations in (16), noisy evaluations can be taken as extra outliers. The justification that \( f^{[1]}(\zeta) \) and/or \( g^{[1]}(\zeta) \) is separated from \( 0 \) for (almost) all \( \ell \neq \lambda_k \) is from [17, Section 3, Lemma 3.1]. As in [17], we use the same justification for correctly identifying non-zero terms via \( \epsilon_{\text{coeff}} \), but here an incorrectly dropped term cannot be reintroduced later. Therefore \( \epsilon_{\text{coeff}} \) should be tight, and falsely kept terms will be removed later. □

**Remark 3.2.** The arising linear systems can be solved by structured linear solvers, e.g., the coefficient matrix in (17) is that in [21, Equ. (10)] ; provided \( \beta_\ell \notin [0, \infty) \) for all \( \ell \). However, the values in \( \Gamma_1 \) are deformed by noise. In [17] we have used a structured total least norm (STLN) iteration to compute the optimal deformation of the diagonal of \( \Gamma_1 \) to achieve a rank deficiency of 1. The arising linear systems in the STLN iterations again have structure and are amenable to a displacement rank approach. How to deal with zeros and poles and the STLN iterations using structured solvers has yet to be worked out. □
We now turn to the main task, namely to interpolate \( f_i \) and \( g_i \) when \( f_{i-1} \) and \( g_{i-1} \) are computed. Suppose the actual supports of \( f_{i-1} \) and \( g_{i-1} \) are \( D_{f_{i-1}} \) and \( D_{g_{i-1}} \) (note Assumption 1). In this case, the possible terms in \( f_i, g_i \) are

\[
\bar{D}_{f,i} = \{ x_1^{d_{1,i}} \cdots x_{i-1}^{d_{i-1,i}} x_i^{d_{i,i}} | x_1^{d_{1,i}} \cdots x_{i-1}^{d_{i-1,i}} \in D_{f_{i-1}}, x_i^{d_{i,i}} \in D_{f_{i-1}}^{[0]} \},
\]

\[
\bar{D}_{g,i} = \{ x_1^{e_{1,i}} \cdots x_{i-1}^{e_{i-1,i}} x_i^{e_{i,i}} | x_1^{e_{1,i}} \cdots x_{i-1}^{e_{i-1,i}} \in D_{g_{i-1}}, x_i^{e_{i,i}} \in D_{g_{i-1}}^{[0]} \}.
\]

Described in Remark 2.4, the new variable \( x_i \) is chosen among \( x_1, \ldots, x_n \) such that the terms sets \( D_{f,i}, D_{g,i} \) in (4) for \( f_i \) and \( g_i \) are as small as possible. We designate the possible terms in \( f_i \) and \( g_i \), represented as (4) and (5), as

\[
D_{f,i} = \{ x_1^{d_{1,i}} \cdots x_{i}^{d_{i,i}} | j = 1, 2, \ldots, l_i/E \} \quad \text{and} \quad D_{g,i} = \{ x_1^{e_{1,i}} \cdots x_{i}^{e_{i,i}} | m = 1, 2, \ldots, l_g/E \}.
\]

The unknown polynomials \( f_i \) and \( g_i \) are represented as

\[
f_i = \sum_{j=1}^{l_i/E} y_{j,i} x_1^{d_{1,i}} \cdots x_i^{d_{i,i}} \quad \text{and} \quad g_i = \sum_{m=1}^{l_g/E} z_1 x_1^{e_{1,i}} \cdots x_i^{e_{i,i}}
\]

where \( y_{j,i} \) and \( z_1 \) are indeterminates.

Let \( b_1, \ldots, b_{l_i/E} \in \mathbb{Z} > 0 \) be sufficient large distinct prime numbers and \( s_j \) be random integers with \( 1 \leq s_j < b_j \). We choose \( \gamma_j = \exp(2\pi i/b_j)^{s_j} \in \mathbb{C} \) for \( 1 \leq j \leq i \) (cf. [9]). In the exact case, discussed in Section 2 above, we know that the dimension of the nullspace of (7) is \( 1 + E - k \) for \( L \geq f_{i,E} g_{i,E} \) evaluations. In fact, \( f_{i,E} g_{i,E} \) is an upper bound which guarantees that the dimension of the nullspace of (7) is \( 1 + E - k \). For the random examples shown in Table 1 and Table 2, our algorithm only needs \( L = f_{i,E} + g_{i,E} + 10 \) probes to obtain \( f_i \) and \( g_i \). In the noisy case, we start from the approximate evaluations for \( \ell = 0, \ldots, L - 1 \),

\[
\beta_{\ell} \approx \gamma_{\ell,E} \gamma_{\ell,E}^* \text{ with } \gamma_{\ell,E} = f(\xi; \ldots; \xi) / g(\xi; \ldots; \xi),
\]

where \( \gamma_{\ell,E} \) is noise or an outlier error. With \( y_{j,i} \) and \( z_1 \) unknown, (20) yield the following linear system:

\[
G \begin{bmatrix} y^T \\ z^T \end{bmatrix} = [V_i, L(\xi_1, \ldots, \xi_n), -V_i, L(\xi_1, \ldots, \xi_n)] \begin{bmatrix} y^T \\ z^T \end{bmatrix} = 0
\]

(c.f. (7)), where \( L = f_{i,E} + g_{i,E} + L_0 \) with \( L_0 \geq 1 \) constant, \( V_i, L(\xi_1, \ldots, \xi_n) \) are Vandermonde matrices, and \( V_i, L(\xi_1, \ldots, \xi_n) \) is \( \text{diag} (0, \ldots, 0) \). One may estimate the numerical rank deficiency of \( G \), denoted by \( \rho \), by computing its SVD. In consequence, the actual count of errors \( k < 1 + E - \rho \) is obtained.


In the situation, the remaining problem can be transformed as the problem of interpolating \( f_i \) and \( g_i \) from their possible terms \( D_{f,i,E} \) and \( D_{g,i,E} \). One algorithm presented in [17], for interpolating sparse rational functions from noisy values, is applied to obtain \( f_i \) and \( g_i \). More details will be found in [17].

**Algorithm Numerical Interpolation of Rational Functions with Outlier Errors**

**Input:** \( \{(x_1, \ldots, x_n) \mid f/g \} \subset \mathbb{C} \) (input as a black box with noise and outlier errors, the latter at a given rate (see Remark 1.1)).

1. Initialize the anchor points and the support of \( f \) and \( g \): choose \( \alpha_1, \alpha_2, \ldots, \alpha_n \) as random roots of unity, let \( D_{f,0} = \{1\} \) and \( D_{g,0} = \{1\} \).

2. For \( i = 1, 2, \ldots, n \) do:
   - Interpolate the univariate polynomials \( f[1] \) and \( g[1] \) and get their supports \( D_{f[1]} \) and \( D_{g[1]} \):
     - Choose a random root of unity \( \xi \) and get the evaluations \( \beta_{\ell}, \xi \) with the noise and the outlier errors as (13).
     - Construct the matrix \( G \) in (15) from \( \beta_{\ell}, \xi \) and \( \xi \). Compute the SVD of \( G \) and find its numerical rank deficiency \( r \). A relative tolerance \( \varepsilon_{\text{rank}} \) for a jump in the singular values can be provided as an additional input.
   - Get the matrix \( \tilde{G} \) from the reduced linear system (16) with \( r \), and then obtain \( f[1] \) and \( g[1] \) from the last singular vector of \( \tilde{G} \).
   - Get the error locator polynomial \( \lambda_i \) by checking (22).
   - Obtain \( f[1] \) and \( g[1] \) by applying univariate polynomial division, and then get the actual support \( D_{f[1]} \) \( f[1] \) by rounding coefficients that are absolutely \( \leq \varepsilon_{\text{coeff}} \) to 0.
   - Similarly, get the actual support \( D_{g[1]} \) \( g[1] \).
   - Let \( D_{f,i} = D_{f[1]} \) and \( D_{g,i} = D_{g[1]} \). For \( i = 2, \ldots, n \) do:
     - Interpolate the polynomials \( f_i \) and \( g_i \) as follows:
       - Choose the variable \( x_i \) from \( x_1, \ldots, x_i \) such that \( D_{f,i,E} \) and \( D_{g,i,E} \) have the fewest elements.
       - Compute \( f_i \) and \( g_i \).
     - (b.1) Choose random roots of unity \( \xi_1, \ldots, \xi_n \). For \( \ell = 0, 1, 2, \ldots \) compute the evaluations \( \beta_{\ell} \) with the noise and the outlier errors as (20).
     - (b.2) Construct the matrix \( G \) in (21) from \( \beta_{\ell} \) and \( D_{f,i,E} \), \( D_{g,i,E} \).
     - (b.3) Compute the SVD of \( G \) and get the actual count of errors \( k \) from the numerical rank deficiency of \( G \).
Reconstruct the possible terms $D_{f,i,k}, D_{g,i,k}$ in $f_{i,k}$ and $g_{i,k}$.

(5) Get the shrink matrix $\tilde{G}$ from $\beta_{i,k}$ and $D_{f,i,k}, D_{g,i,k}$.

(6) Obtain $f_{i,k}, g_{i,k}$ from the last singular vector of $\tilde{G}$.

(7) Obtain $f_i$ and $g_i$ by the structured total least norm technique presented in [17], and then get their actual supports $D_{f,i}, D_{g,i}$.

4. With the support of $f_n$ and $g_n$, interpolate $f(x_1, \ldots, x_n)/c$ and $g(x_1, \ldots, x_n)/c$ again to improve the accuracy of the coefficients:

(a) Construct the linear system from the approximate $\beta_{n,t}$ as (23) and the exact terms $D_{f,n}$ and $D_{g,n}$.

(b) Compute the refined solution $\tilde{y}$ and $\tilde{z}$ by use of STLN method.

(c) Obtain $f(x_1, \ldots, x_n)/c$ and $g(x_1, \ldots, x_n)/c$ from $\tilde{y}, \tilde{z}$ and $D_{f,n}, D_{g,n}$.

4. EXPERIMENTS

Our algorithm has been implemented in Maple and the performance is reported in the following three tables. All examples in Table 1 and Table 2 are run in Maple 15 under Windows for Digital=15. In Table 1 we exhibit the performance of our algorithm for recovering univariate rational functions from a black box that returns noisy values with outlier errors. For each example, we construct two relatively prime polynomials with random integer coefficients in the range $-5 \leq c \leq 5$. Here Random Noise denotes the range of relative noise randomly added to the black box evaluations of $f/g$: $\tilde{d}_j \geq \deg(f)$ and $\tilde{g}_k \geq \deg(g)$ denote the degree bound of the numerator and the denominator, respectively; $t_f$ and $t_g$ denote the number of terms of the numerator and denominator, respectively; $1/q$ denotes the error rate of the outlier error; Rel. Error is the relative error, namely $$\frac{(||f - f||_2 + ||g - g||_2)}{(||f||_2 + ||g||_2)}.$$ where $\bar{f}/\bar{g}$ is the fraction computed by our algorithm and $c$ is optimally chosen to minimize the error. For each example, the outlier error is the relative error of the evaluation, which is in the range of 0.01 × [100, 200] or 0.01 × [200, 300]. Running times serve to give a rough idea on the efficiency, and are for SONY VAIO laptops with 8GB of memory and 2.67GHz and 2.80GHz Intel I7 processors.

<table>
<thead>
<tr>
<th>Ex.</th>
<th>Random Noise</th>
<th>$d_f, d_g$</th>
<th>$\deg(f), \deg(g)$</th>
<th>$t_f, t_g$</th>
<th>$1/q$</th>
<th>Time (sec.)</th>
<th>Rel. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$10^{-4}$ - $10^{-2}$</td>
<td>10, 10</td>
<td>3, 3</td>
<td>3, 1</td>
<td>0.3</td>
<td>37.5</td>
<td>6.0e-7</td>
</tr>
<tr>
<td>2</td>
<td>$10^{-5}$ - $10^{-3}$</td>
<td>6, 6</td>
<td>4, 5</td>
<td>2, 4</td>
<td>0.39</td>
<td>4.4</td>
<td>3.1e-6</td>
</tr>
<tr>
<td>3</td>
<td>$10^{-6}$ - $10^{-4}$</td>
<td>18, 13</td>
<td>8, 3</td>
<td>4, 3</td>
<td>0.26</td>
<td>2.5</td>
<td>2.3e-8</td>
</tr>
<tr>
<td>4</td>
<td>$10^{-7}$ - $10^{-5}$</td>
<td>20, 20</td>
<td>10, 10</td>
<td>4, 4</td>
<td>0.32</td>
<td>8.4</td>
<td>2.6e-4</td>
</tr>
<tr>
<td>5</td>
<td>$10^{-8}$ - $10^{-6}$</td>
<td>18, 30</td>
<td>3, 15</td>
<td>2, 6</td>
<td>0.3</td>
<td>9.9</td>
<td>8.8e-8</td>
</tr>
<tr>
<td>6</td>
<td>$10^{-9}$ - $10^{-7}$</td>
<td>40, 40</td>
<td>20, 20</td>
<td>5, 5</td>
<td>0.48</td>
<td>32</td>
<td>6.2e-9</td>
</tr>
<tr>
<td>7</td>
<td>$10^{-10}$ - $10^{-8}$</td>
<td>50, 30</td>
<td>30, 7</td>
<td>6, 3</td>
<td>0.34</td>
<td>24</td>
<td>8.6e-6</td>
</tr>
<tr>
<td>8</td>
<td>$10^{-11}$ - $10^{-9}$</td>
<td>30, 70</td>
<td>5, 40</td>
<td>4, 7</td>
<td>0.61</td>
<td>57</td>
<td>2.7e-9</td>
</tr>
<tr>
<td>9</td>
<td>$10^{-12}$ - $10^{-10}$</td>
<td>80, 80</td>
<td>50, 50</td>
<td>5, 5</td>
<td>0.52</td>
<td>29</td>
<td>7.2e-10</td>
</tr>
<tr>
<td>10</td>
<td>$10^{-13}$ - $10^{-11}$</td>
<td>80, 80</td>
<td>50, 50</td>
<td>51, 51</td>
<td>1.8</td>
<td>31</td>
<td>3.2e-7</td>
</tr>
</tbody>
</table>

Table 1: Performance: univariate case

Remark 4.1. In our tests, the $k$ outlier errors are introduced in random locations after the bound $G \geq k$ is derived from the error rate $1/q$. However, our algorithm also makes random choices, the anchor points $a$ and the random roots of unity $\zeta$ (see Step 2(a)). We perform 20 trials of the $\zeta$’s before giving up with recovery. Running times can fluctuate as a new random choice for $\zeta$ has a new number of outlier errors $k$ in different places. For instance, Example 8 in Table 1 is a case where 10 trials are needed before success. Because our error rates are quite large, the tests cannot succeed simply because the batches have few outlier errors (see the column for $E$ in Table 1). Example 10 in Table 1 is a dense rational function. Our algorithm currently fails to recover the fraction with an error rate of $1/q = 1/4$. □

In Table 2 we exhibit the performance of our algorithm on multivariate inputs. For each example, we construct two relatively prime multivariate polynomials with random integer coefficients in the range $-5 \leq c \leq 5$. Here Random Noise denotes the noise in this range randomly added to the black box of $f/g$: $\tilde{d}_j \geq \deg(f)$ and $\tilde{d}_g \geq \deg(g)$ denote the degree bound of the numerator and the denominator, respectively; $t_f$ and $t_g$ denote the number of terms of the numerator and denominator, respectively; $n$ denotes the number of variables of the rational functions; $N$ denotes the number of the black box probes needed to interpolate the approximate multivariate rational function; $E$ denotes the maximum number of outliers for each individual interpolation step, $1/q$ the resulting error rate of the outlier error; finally, Rel. Error denotes the relative backward error computed by our algorithm. About the setting of the outlier error, it is the same as the univariate case. Example 13 is one polynomial test which shows that our algorithm can also deal with the multivariate polynomial interpolation from values with noise and outlier errors.

Example 14 in Table 2 warrants further discussion, as the number of probes for a fraction with 5 terms in both the numerator and denominator takes over 10000 evaluations. There are $n = 102$ variables. Estimating the degree in each variable, using as upper bounds $d_f$ and $d_g$, consumes $(d_f + d_g + 2E + L_0)$-102 probes, about 5000. We then use $x_i$ as the variable in the error locator polynomial $A_i$; see Remark 2.3. We have for each $i$ the estimates $|D_{i,k,\tilde{E}}| \approx t_f(\deg_d(f) + 1 + E) = 5(3 + 1 + 1)$ and $|D_{i,k,\tilde{E}}| = t_g(\deg_d(g) + 1 + E) = 5(2 + 1 + 1)$, that is $45 + 102$ evaluations, or about 5000 in total. Using sharper upper bounds for $\deg_d(f)$ and $\deg_d(g)$ one could reduce the number of probes to about 6000. The fact remains that 102 variables constitute a large recovery problem, with $(5 + 5) \times 102$ individual exponents $d_{j,\mu,\nu}$ to be determined.
We have an objective value of 6. The produced fraction can be recovered by oversampling at $\beta$. We compute
\[ \sum_{\ell=1}^{L} g_\ell(\mathbf{x}) = 12, \]
and $\mathbf{y} = (f_\ell(\mathbf{x}))$ for a real vector $[\mathbf{y}, \mathbf{z}]^T \in \mathbb{R}^2$ with the following constraint: $y_{21} = 1$, meaning the numerator polynomial $f$ is monic of degree 12. By separating real and imaginary parts of the matrix $G$ in (15) we have the linear equational constraints
\[ [\text{Realpart}(V_1), -\text{Realpart}(\Pi W_1)] [\mathbf{y}, \mathbf{z}]^T = [0, y_{21}=1]. \]

The linear system (25) has a higher dimensional solution set because $f$ and $g$ are not relatively prime. We wish to discover a sparse solution by minimizing $\sum_{i=1}^m |y_i| + \sum_{i=1}^n |z_i| = \|\mathbf{y}, \mathbf{z}\|$ via Tshelyshyev's linear programming formulation.

We solve the Maple code to Optimization['LPsolve'] with method = active set produces the solution $f = x_2^{12} - 1.0 x_1^{11} - 1.78 \times 10^{-10} x_{10}^{10} + 1.72 \times 10^{-10} x_9^9 + 1.72 \times 10^{-10} x_8^8 - 1.78 \times 10^{-10} x_7^6 + 1.82 \times 10^{-10} x_6^5 - 1.88 \times 10^{-10} x_5^4 + 1.92 \times 10^{-10} x_4^3 - 1.88 \times 10^{-10} x_3^2 + 1.0 x - 1.0$ and $g = 1.0 x^6 + 1.0 x^5 - 5.83 \times 10^{-12} x^3 - 2.07 \times 10^{-12} x^2 - 1.0 x - 1.0$ with an objective value of 6.9999999999878. The rounded polynomials give the unreduced form in (24).

For some examples of lesser unreduced sparsity, the unreduced fraction can be recovered by oversampling at $L_0$ additional values. Without oversampling, the Maple 16 LPsolve call falsely reports infeasibility of the linear program.

We plan to study sparse unreduced recovery in the presence of outlier errors à la [19] and with Zippel lifting to several variables in the near future.

5. REFERENCES

Note added to Remark 2.2: We expand on using fresh evaluation points \((\xi_1, \ldots, \xi_\ell) \in K^\ell\) on each black box evaluation. Let \(v_1, \ldots, v_\ell\) be new indeterminates. Then for the evaluation point \((\xi_1, \ldots, \xi_\ell) \leftarrow (v_1, \ldots, v_\ell)\), Assumption 4 is satisfied over the rational function field \(K_{\ell,v} = K(v_1, \ldots, v_\ell)\). Therefore the solutions to the system of \(L \geq |D_{f,\ell,E} + D_{g,\ell,E}|\) homogeneous linear equations (6) and (7) in \(M = |D_{f,\ell,E} + D_{g,\ell,E}|\) unknowns over \(K_{\ell,v}\) yield polynomials \(f\) and \(g\) that satisfy (10), namely

\[
f, g - f g_i = 0, \quad f \neq 0, g \neq 0 \quad \text{or} \quad f = g = 0.
\]

We remark that the black box for \(f/g\) is not assumed to evaluate at points in \(K_{\ell,v}\), here we use such evaluations only for purpose of proof. Now if instead of \((v^f_1, \ldots, v^f_\ell)\) for each \(\ell\) one were to evaluate at \((v^g_1, \ldots, v^g_\ell)\), that over the new function field \(K_{\ell,g,v} = K(v^g_1, \ldots, v^g_\ell, v_{\ell+1}, \ldots, v_{\ell+\ell-1})\), we will argue that (26) remains valid for solution polynomials \(f\) and \(g\) over \(K_{\ell,v}\). As above, we get the property corresponding to (9), namely

\[
\forall \ell, 0 < \ell \leq L - 1 : \quad (f^g A^q)(v^g_1, \ldots, v^g_\ell) = (f^g g^A)(v^g_1, \ldots, v^g_\ell).
\]

Note that there are no true poles, and the \(\beta_i, \lambda_\ell\) can be anything. Now the coefficient vector of \(f A^q g - f g A^q\) is a nullspace vector to a non-singular matrix. Non-singularity can be shown by evaluating \(v^g_i, v^g_i \leftarrow v^f_i\) to obtain a transposed Vandermonde matrix over \(K_{\ell,v}\), or by showing that in the minor expansion the lexicographically largest term cannot be canceled. By the Zippel-Schwartz Lemma non-singularity is preserved for randomly and uniformly sampled \(\xi_\ell \in S \subseteq K\), that independently of the error locations \(\lambda_\ell\) or false evaluations \(\beta_i\). At a true pole we have \(g(A\xi_\ell) = 0\) which yields (27) for that \(\ell\) at the specialization \((v^g_1, \ldots, v^g_\ell) \leftarrow (\xi^g_1, \ldots, \xi^g_\ell)\).

We can provably reduce the number of evaluations at random points \((\xi^g_1, \ldots, \xi^g_\ell) \in S^\ell\) to \(L = |D_{f,\ell,E} + D_{g,\ell,E}| + E - 1 = M + E - 1\). We consider the space of polynomials \((f, g)\) corresponding to the solutions \([g, \bar{z}]^T\) of

\[
\sum_{m,\nu,\eta} z_{m,\nu,\eta} \xi_{m,\nu,\eta}^{m+1} + v_{m,\nu} \xi_{m,\nu,\eta}^{m+2} \cdots \xi_{m+1,\nu,\eta}^{m+1 - \nu} \xi_{m,\nu,\eta}^{m+1} = 0
\]

if \(\beta_i \ell = \infty\). (29)

Our convention was that for \(g_i(\xi_1, \ldots, \xi_\ell) = 0\) we set \(\gamma_i, \ell = \infty\) even if \(f_i(\xi_1, \ldots, \xi_\ell) = 0\) (sentence below Assumption 2). We claim that the solutions \((f, g)\) of (26), namely \(f g - f g = 0\), with the maximal term sets prescribed by (4) and (5) and with the additional linear homogeneous constraints

\[
\bar{g}(\xi_1, \ldots, \xi_\ell, \lambda_\ell) = \beta_i, \lambda_\ell g(\xi_1, \ldots, \xi_\ell) \quad \text{for all} \quad 1 \leq \kappa \leq k, \quad \text{using} \quad \bar{g}(\xi_1, \ldots, \xi_\ell, \lambda_\ell) = 0 \quad \text{for} \quad \beta_i, \lambda_\ell = \infty,
\]

form a subspace of the former, corresponding to (28), (29). The claim for \(\ell \neq (\lambda_1, \ldots, \lambda_\ell)\) follows from \(\beta_i, \ell g(g, \bar{y})(\xi_1, \ldots, \xi_\ell) = (f_i, \ell g) g(\xi_1, \ldots, \xi_\ell)\) (by the definition of \(\beta_i, \ell\)) (by (26)). Now if \(g_i(\xi_1, \ldots, \xi_\ell) \neq 0\) we have (28). For \(g_i(\xi_1, \ldots, \xi_\ell) = 0\) we have \((f_i, \ell g)(\xi_1, \ldots, \xi_\ell) = 0\), and if \(f_i(\xi_1, \ldots, \xi_\ell) \neq 0\) we obtain the corresponding (29); otherwise we use (31).

We seek a condition on the \(\xi_\ell \in K\) so that the two vectors spaces are the same. First, we consider the case \(k = 0\): for symbolic values \(\xi_\ell \leftarrow v_\ell\) we have shown above that with \([D_{f,\ell,E} + D_{g,\ell,E}]\) many equations (28) we get (26). There are \(M = |D_{f,\ell,E} + D_{g,\ell,E}|\) variables, so we can select \(M - 1\) equations (“row subsetting”) that preserve the rank of the coefficient matrix and therefore still have the same solution space. Note that there exists at least one non-zero solution, the pair \((f_i, g_i)\), and that \(f_i/g_i\) may be unreduced. The constraint (30) is vacuous because \(k = 0\) and the constraint (31) is vacuous because there are no true poles or zeros at symbolic evaluations. All equations are formed by substituting new variables \(v_\ell\) for \(x_\ell\), so the rank is also preserved by the first \(M - 1\) equations. Note that the new coefficient matrix is over the smaller rational function field \(K(v_{l+1}, \ldots, v_{\ell+\ell-1})\), but the rank remains invariant over additional transcendental extensions. Now any specialization \((v^g_1, \ldots, v^g_\ell) \leftarrow (\xi^g_1, \ldots, \xi^g_\ell) \in S^\ell\) for \(0 \leq \ell \leq M - 2\) that preserves the (symbolic) rank and \(2. \) does not contain a (true) pole, that is \(g_i(\xi^g_1, \ldots, \xi^g_\ell) \neq 0\), will force (26). The solution space \((f, g)\) of (26) could hypothetically have a larger dimension at the specialization, in which case it no longer would be a subspace of (28), an impossibility. Again (29), (30) and (31) are vacuous: \(k = 0\) and there are no poles. Note that we actually used \(M + E - 1\) equations, which increases the probability of preserving the rank of the coefficient matrix of (28), (29) in the presence of \(k = 0\) errors. We remark that if \(D_{f,\ell,E} = D_{g,\ell,E}\) then \(V_{\ell,L} = W_{\ell,L}\) in (7) and the higher rank can only be achieved by \(\Gamma_{\ell,\ell,\ell}\) containing specific values, namely evaluations of \(f_i/g_i\).

The case \(1 \leq k \leq E\) is based on the case \(k = 0\). There are \(M - 1\) equations (28) with correct \(\beta_i, \ell = (f_i/g_i)(\xi_1, \ldots, \xi_\ell)\). For those equations, we want our specializations to satisfy the above conditions of the case \(k = 0\), which imply that all their solution polynomials satisfy (26). Note that the black box can produce erroneous equations adaptively to the evaluation points \((\xi_1, \ldots, \xi_\ell)\), akin to the adaptive cipher text attack in public key cryptosystems. However, the black box can do this at most \(E\) times, and has no
control over the remaining $M - 1$ equations. Our evaluations are random for each $\ell$, so they are random also for those (unknown) $M - 1$ equations. The remaining $k$ equations with erroneous $\beta_{i,\lambda}$ simply restrict the solution to $f(\xi_{1,\lambda}, \ldots) = \bar{g}(\xi_{1,\lambda}, \ldots) = 0$ (see also explanation below (11)) and (26) remains valid. The remaining $E - k$ equations for true evaluations are satisfied by (26), that even at true poles. Again, there is at least one non-zero solution $(f_i, \Lambda_i, g_i, \Lambda_i)$.

Using the Zippel-Schwartz Lemma, the probability that all solutions $f, \bar{g}$ derived from (28, 29) for $0 \leq \ell \leq M + E - 2 = |D_{f,1,\ell}| + |D_{g,1,\ell}| + E - 2$ satisfy (26), that is, $f, \bar{g} = f_i, g_i = 0$, when selecting $\xi_{\mu,\ell} \in S \subseteq K$ uniformly and randomly, is bounded from below as

$$\geq \left(1 - \frac{\deg(g_i)}{|S|}\right)^{M - 1} \left(1 - \frac{(M - 1)(\deg(f_i) + \deg(g_i))}{|S|}\right) \geq 1 - \frac{(M - 1)(\deg(f_i) + 2 \deg(g_i))}{|S|}.$$ 

For such random $\xi_{\mu,\ell}$ the provably worst case number of evaluations of Remark 2.3 is reduced, and the error rate of Remark 2.6 is provably achieved, as well.

\textit{Erich Kaltofen, July 14, 2013} □

<table>
<thead>
<tr>
<th>Notation (in alphabetic order):</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$ the anchor point argument values for variables</td>
</tr>
<tr>
<td>$\beta$ the possibly erroneous values returned by the black box for $f/g$</td>
</tr>
<tr>
<td>$\gamma$ the correct evaluations for $f/g$</td>
</tr>
<tr>
<td>$d_{j,\mu}$ the degree of variable $x_{\mu}$ in the $j$-th term in $f$</td>
</tr>
<tr>
<td>$d_f \geq \deg(f)$, a bound that is input</td>
</tr>
<tr>
<td>$d_g \geq \deg(g)$, a bound that is input</td>
</tr>
<tr>
<td>$D$ sets of terms (non-zero monomials)</td>
</tr>
<tr>
<td>$e_{m,\mu}$ the degree of variable $x_{\mu}$ in the $m$-th term in $g$</td>
</tr>
<tr>
<td>$\epsilon$ a numerical tolerance for algorithmic decisions; not an outlier error, as is sometimes used</td>
</tr>
<tr>
<td>$E \geq k$, an upper bound on the number of errors that is input to the algorithm</td>
</tr>
<tr>
<td>$f$ the numerator polynomial</td>
</tr>
<tr>
<td>$g$ the denominator polynomial</td>
</tr>
<tr>
<td>$k$ the actual number of errors, to be determined by the algorithms</td>
</tr>
<tr>
<td>$K$ an arbitrary field with exact arithmetic</td>
</tr>
<tr>
<td>$L$ the length of the list of a batch of evaluations</td>
</tr>
<tr>
<td>$\lambda_{\kappa}$ $1 \leq \kappa \leq k$, are the positions of the erroneous evaluations in the list of evaluations</td>
</tr>
<tr>
<td>$\Lambda$ the error locator polynomial (3); see see Remarks 1.1 and 2.6</td>
</tr>
<tr>
<td>$M$ the number of unknowns in our linear systems</td>
</tr>
<tr>
<td>$n$ is the number of variables</td>
</tr>
<tr>
<td>$N$ the needed number of black box evaluations</td>
</tr>
<tr>
<td>$q$ $1/q$ is the error rate; see Remarks 1.1 and 2.6</td>
</tr>
<tr>
<td>$S$ a finite set from which elements are randomly sampled</td>
</tr>
<tr>
<td>$\sigma$ a scalar shift value; see Remark 2.8</td>
</tr>
<tr>
<td>$t$ denotes the actual number of terms</td>
</tr>
<tr>
<td>$T \geq t$, an upper bound that is input</td>
</tr>
<tr>
<td>$\Theta$ a lower bound for the relative outlier errors</td>
</tr>
<tr>
<td>$u, v$ auxiliary indeterminates</td>
</tr>
<tr>
<td>$x_1, \ldots, x_n$ the variables of $f/g$</td>
</tr>
<tr>
<td>$\xi$ values for the variables from a field $\in K$</td>
</tr>
<tr>
<td>$\bar{y}$ the unknown coefficient vector of the numerator $f$</td>
</tr>
<tr>
<td>$\bar{z}$ the unknown coefficient vector of the denominator $g$</td>
</tr>
<tr>
<td>$\zeta$ values for the variables $\in \mathbb{C}$</td>
</tr>
</tbody>
</table>