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 $k \le E$ 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 $T \ge t$ and $E \ge k$ 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 $t \le T$ 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

*This research was supported in part by the National Science Foundation under Grant CCF-1115772 (Kaltofen).

ISSAC'13, June 26–29, 2013, Boston, Massachusetts, USA.

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., $\deg(f) \le d-1$ and k errors, one needs $L = d + 2E$ evaluations, where $E \ge k$ bounds the number of faults a-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 $L \ge L_{\min}$ evaluations produces faulty values for no more than $k \le L/q$ inputs, where $q > 2$. Here $1/q$ is the error rate, and $L_{\min}$ 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 $L/q = (d+2E)/q \le (d+2d/(q-2))/q = d/(q-2)$, so $k \le E$ 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 $(f/\mathrm{GCD}(f,g))/(g/\mathrm{GCD}(f,g))$. 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, Section 4.1]. Consider the rational function $f/g \in \mathsf{K}(x_1, \ldots, x_n)$, where the numerator and denominator are represented as

$$f = \sum_{j=1}^{t_f} a_j \vec{x}^{\,\vec{d}_j},\ g = \sum_{m=1}^{t_g} b_m \vec{x}^{\,\vec{e}_m},\ a_j, b_m \in \mathsf{K} \setminus \{0\}, \qquad (1)$$

where K is an arbitrary field and the terms are denoted by $\vec{x}^{\,\vec{d}_j} = x_1^{d_{j,1}} \cdots x_n^{d_{j,n}}$ and $\vec{x}^{\,\vec{e}_m} = x_1^{e_{m,1}} \cdots x_n^{e_{m,n}}$. We analyze our variant of Zippel's sparse interpolation technique to recover the numerator and denominator. Zippel's technique [12, Section 4] determines the support of $f_i = f(x_1, \ldots, x_i, \alpha_{i+1}, \ldots, \alpha_n)$ and $g_i = g(x_1, \ldots, x_i, \alpha_{i+1}, \ldots, \alpha_n)$ iteratively from the support of f_{i-1} and g_{i-1}, where $\alpha_2, \ldots, \alpha_n \in \mathsf{K}$ is a random anchor point. We will use Zippel's probabilistic assumption.

Assumption 1. *Each term $x_1^{d_{j,1}} \cdots x_{i-1}^{d_{j,i-1}}$, where $1 \le j \le t_f$, and each term $x_1^{e_{m,1}} \cdots x_{i-1}^{e_{m,i-1}}$, where $1 \le m \le t_g$, has a non-zero coefficient in f_{i-1} and g_{i-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_i and g_i from the sparse supports of f_{i-1} and g_{i-1} and evaluations of $f_i(x_1, \ldots, x_i)/\, g_i(x_1, \ldots, x_i) = f(x_1, \ldots, x_i, \alpha_{i+1}, \ldots, \alpha_n)/\, g(x_1, \ldots, x_i, \alpha_{i+1}, \ldots, \alpha_n)$. We chose $\xi_1, \ldots, \xi_i \in \mathsf{K}$ and evaluate at powers $\xi_1^\ell, \ldots, \xi_i^\ell$, where $\ell = 0, 1, 2, \ldots$ We will obtain

$$\beta_{i,\ell} = \gamma_{i,\ell} + \gamma'_{i,\ell},\ \text{where } \gamma_{i,\ell} = \frac{f_i(\xi_1^\ell, \ldots, \xi_i^\ell)}{g_i(\xi_1^\ell, \ldots, \xi_i^\ell)} \in \mathsf{K} \cup \{\infty\} \qquad (2)$$

and where $\gamma'_{i,\lambda_\kappa} \ne 0$ exactly at the $k \le E$ unknown indices $0 \le \lambda_1 < \lambda_2 < \cdots < \lambda_k$ for ℓ, 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 λ_κ.*

If $g_i(\xi_1^\ell, \ldots, \xi_i^\ell) = 0$ we have $\gamma_{i,\ell} = \infty$, but $\beta_{i,\ell}$ can be erroneously $\in \mathsf{K}$. Similarly, if $g_i(\xi_1^\ell, \ldots, \xi_i^\ell) \ne 0$ we may erroneously have $\beta_{i,\ell} = \infty$.

Following the Berlekamp/Welch strategy, we attempt to recover $(f_i \Lambda_i)/(g_i \Lambda_i)$ where

$$\Lambda_i = (x_1 - \xi_1^{\lambda_1}) \cdots (x_1 - \xi_1^{\lambda_k}) \qquad (3)$$

is an error locator polynomial. The set of possible terms in $f_i \Lambda_i$ and $g_i \Lambda_i$ can now be restricted to

$$D_{f,i,E} = \{x_1^{d_{j,1}+\nu} x_2^{d_{j,2}} \cdots x_{i-1}^{d_{j,i-1}} x_i^{\delta_j} \mid 1 \le j \le t_f, 0 \le \nu \le E,$$
$$0 \le \delta_j \le \min(\deg(f) - d_{j,1} - \cdots - d_{j,i-1}, \deg_{x_i}(f))\}, \qquad (4)$$

$$D_{g,i,E} = \{x_1^{e_{m,1}+\nu} x_2^{e_{m,2}} \cdots x_{i-1}^{e_{m,i-1}} x_i^{\eta_m} \mid 1 \le m \le t_g, 0 \le \nu \le E,$$
$$0 \le \eta_m \le \min(\deg(g) - e_{m,1} - \cdots - e_{m,i-1}, \deg_{x_i}(g))\}. \qquad (5)$$

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_{i-1} and g_{i-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* $\deg(f)$, $\deg_{x_i}(f)$, $\deg(g)$ *and* $\deg_{x_i}(g)$.

Let $\vec{y}$ and $\vec{z}$ be the coefficient vectors of $f_i \Lambda_i$ and $g_i \Lambda_i$ for the (distinct) terms in $D_{f,i,E}$ and $D_{g,i,E}$. For any $\ell = 0, 1, 2, \ldots$

and any point $\xi_1, \ldots, \xi_i \in \mathsf{K}$ each value $\beta_{i,\ell}$ in (2) constitutes a linear equation for the coefficient vector,

$$\sum_{j,\nu,\delta} y_{j,\nu,\delta}(\xi_1^{d_{j,1}+\nu}\xi_2^{d_{j,2}}\cdots\xi_{i-1}^{d_{j,i-1}}\xi_i^{\delta})^\ell = \beta_{i,\ell}\sum_{m,\nu,\eta} z_{m,\nu,\eta}(\xi_1^{e_{m,1}+\nu}\xi_2^{e_{m,2}}\cdots\xi_{i-1}^{e_{m,i-1}}\xi_i^{\eta})^\ell. \quad (6)$$

Errors in the β_{i,λ_κ} are tolerated because $f_i\Lambda_i$ and $g_i\Lambda_i$ are both $= 0$ at $x_1 = \xi_1^{\lambda_\kappa}$. Note again that coefficients/indeterminates can be the same, $y_{j_1,\nu_1,\delta} = y_{j_2,\nu_2,\delta}$, 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 $y_{j,\nu,\delta}$ and $z_{m,\nu,\eta}$,

$$V_{i,L}(\xi_1, \ldots, \xi_i)\vec{\boldsymbol{y}}^T = \Gamma_{i,L}W_{i,L}(\xi_1, \ldots, \xi_i)\vec{\boldsymbol{z}}^T, \quad (7)$$

where $\Gamma_{i,L}$ is a diagonal matrix of rational function values $\beta_{i,\ell}$ and $V_{i,L}$ and $W_{i,L}$ are (transposed) Vandermonde matrices, with possible zero rows. If $\beta_{i,\ell} = \infty$ then the ℓ-th row in $V_{i,L}$ is set to a zero row, and $(\Gamma_{i,L})_{\ell,\ell}$ 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 vector $[\vec{\boldsymbol{y}}, \vec{\boldsymbol{z}}]$ of $f_i\Lambda_i$ and $g_i\Lambda_i$ solves (7). For the term sets in (4) and (5) let

$$D_{f,i,E}D_{g,i,E} = \{\tau_f \cdot \tau_g \mid \tau_f \in D_{f,i,E}, \tau_g \in D_{g,i,E}\}, \quad (8)$$

with $|D_{f,i,E}D_{g,i,E}| \le |D_{f,i,E}| \cdot |D_{g,i,E}|$. Now set $L \ge |D_{f,i,E} \times D_{g,i,E}|$ in (7). We argue that for random $\xi_1, \ldots, \xi_i$, the polynomials $\bar{f}$ and $\bar{g}$, which correspond to any non-zero solution vector $[\vec{\boldsymbol{y}}, \vec{\boldsymbol{z}}]$, respectively, of the linear system (7), with high probability satisfy $\bar{f}g_i\Lambda_i = \bar{g}f_i\Lambda_i$.

We shall first assume that the random choices for $\xi_1, \ldots, \xi_i \in S \subset \mathsf{K}$ are such that no two distinct terms in $D_{f,i,E}$ and no two distinct terms in $D_{g,i,E}$ evaluate at $x_\mu \leftarrow \xi_\mu$, $1 \le \mu \le i$, to the same element in K. Because

$$\forall \ell, 0 \le \ell \le L-1: \begin{cases} (f_i\Lambda_i)(\xi_1^\ell, \ldots, \xi_i^\ell) = \beta_{i,\ell}\ (g_i\Lambda_i)(\xi_1^\ell, \ldots, \xi_i^\ell), \\ \bar{f}(\xi_1^\ell, \ldots, \xi_i^\ell) = \beta_{i,\ell} \quad \bar{g}(\xi_1^\ell, \ldots, \xi_i^\ell), \end{cases}$$

we have

$$\forall \ell, 0 \le \ell \le L-1: (f_i\Lambda_i\bar{g})(\xi_1^\ell, \ldots, \xi_i^\ell) = (\bar{f}g_i\Lambda_i)(\xi_1^\ell, \ldots, \xi_i^\ell). \quad (9)$$

Note that if $\beta_{i,\ell} = 0$ then $(f_i\Lambda_i)(\xi_1^\ell, \ldots, \xi_i^\ell) = \bar{f}(\xi_1^\ell, \ldots, \xi_i^\ell) = 0$ and if $\beta_{i,\ell} = \infty$ then $(g_i\Lambda_i)(\xi_1^\ell, \ldots, \xi_i^\ell) = \bar{g}(\xi_1^\ell, \ldots, \xi_i^\ell) = 0$. The possibly occurring terms of the polynomial $f_i\Lambda_i\ \bar{g} - \bar{f}\ g_i\Lambda_i$ are contained in $D_{f,i,E}D_{g,i,E}$ of (8). Note that for $i = 1$ we have $L = |D_{f,1,E}D_{g,1,E}| = \deg_{x_1}(f) + \deg_{x_1}(g) + 1 + 2E$; see Remark 2.5 below.

Assumption 4. *Finally, we assume that the random choices for $\xi_1, \ldots, \xi_i \in S$ are such that no two terms in $D_{f,i,E}D_{g,i,E}$ evaluate to the same value, which subsumes our earlier assumption on the distinctness of term evaluations.*

For $L \ge |D_{f,i,E}E_{g,i,E}|$ we then must have

$$f_i\Lambda_i\ \bar{g} - \bar{f}\ g_i\Lambda_i = 0, \quad \bar{f} \ne 0, \bar{g} \ne 0, \quad (10)$$

because the coefficient vector of $f_i\Lambda_i\ \bar{g} - \bar{f}\ g_i\Lambda_i$ is by (9) a kernel vector in a square non-singular (transposed) Vandermonde matrix and therefore must be zero. Since $f_i \ne 0$, $g_i \ne 0$ and $\Lambda_i \ne 0$ and $[\vec{\boldsymbol{y}}, \vec{\boldsymbol{z}}] \ne 0$ both $\bar{f}$ and $\bar{g}$ are non-zero. 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 ∞ from the evaluations $\gamma_{i,\ell}$ 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* $\mathsf{K}[x_1, \ldots, x_n]$ *in* (1).

Suppose now the f_i and g_i are relatively prime in $\mathsf{K}[x_1, \ldots, x_i]$. For random anchor points $\alpha_2, \ldots, \alpha_n$, this will be true with high probability. In fact, f_i and g_i are then random projections of the primitive parts of f and g after removing their contents in $\mathsf{K}[x_{i+1}, \ldots, x_n]$; see Remark 2.4. So we obtain from (10)

$$f_i/g_i = \bar{f}/\bar{g}, \quad (11)$$

by removing a common factor $h \in \mathsf{K}[x_1]$ of $\bar{f}$ and $\bar{g}$. Because of the degree constraints in the term supports in (4) and (5), no additional polynomial factors in the variables $x_2, \ldots, x_i$ are possible. We thus have $hf_i = \bar{f}$ and $hg_i = \bar{g}$. All $x_1 - \xi_1^{\lambda_\kappa}$ for all $\kappa = 1, \ldots, k$ divide $h(x_1)$, because by (11) $\bar{f}(\xi_1^{\lambda_\kappa}, \ldots, \xi_i^{\lambda_\kappa}) = \beta_{i,\lambda_\kappa}\bar{g}(\xi_1^{\lambda_\kappa}, \ldots, \xi_i^{\lambda_\kappa})$, but $f_i(\xi_1^{\lambda_\kappa}, \ldots, \xi_i^{\lambda_\kappa}) \ne \beta_{i,\lambda_\kappa}g_i(\xi_1^{\lambda_\kappa}, \ldots, \xi_i^{\lambda_\kappa})$. Because we estimate the number of errors by E in (4) and (5), $\bar{f}$ and $\bar{g}$ can have common factors in $\mathsf{K}[x_1]$ in addition to $\Lambda_i(x_1)$. Those common factors give the nullspace of (9) a dimension $1 + E - k$, and the kernel vectors corresponding to the lowest degree polynomials $\bar{f}$ and $\bar{g}$ in x_1 are the coefficient vectors of $\bar{f} = cf_i\Lambda_i$ and $\bar{g} = cg_i\Lambda_i$ for a $c \in \mathsf{K}$, $c \ne 0$.

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. □

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 $M = |D_{f,i,E}| + |D_{g,i,E}|$ variables, so at least $L \ge M - 1$ equations are needed. In fact, from earlier iterations one has additional linear constraints for $\mu = 1, 2, \ldots, i-1$:

$$\sum_{j,\nu,\delta} y_{j,\nu,\delta}(\xi_1^{d_{j,1}+\nu}\xi_2^{d_{j,2}}\cdots\xi_\mu^{d_{j,\mu}})^\ell\alpha_{\mu+1}^{d_{j,\mu+1}}\cdots\alpha_i^\delta = \gamma_{\mu,\ell} \times \sum_{m,\nu,\eta} z_{m,\nu,\eta}(\xi_1^{e_{m,1}+\nu}\xi_2^{e_{m,2}}\cdots\xi_\mu^{e_{m,\mu}})^\ell\alpha_{\mu+1}^{e_{m,\mu+1}}\cdots\alpha_i^\eta. \quad (12)$$

Note that after f_μ and g_μ have been computed, the errors in the $\beta_{\mu,\lambda_\kappa}$ can be corrected.

If $k < E$, one may simultaneously grow $E = k = 0, 1, \ldots$ in $D_{f,i,E}$ and $D_{g,i,E}$ (see (4) and (5)), that is, add new columns to (7). The objective is to produce a single non-zero solution $\bar{f}$ and $\bar{g}$ with the property that the evaluations have k errors located by $h(x_1) = \Lambda_i(x_1)$. The constraints (12) guarantee that the corresponding $f_i = \bar{f}/\Lambda_i$ and $g_i = \bar{g}/\Lambda_i$ project to f_μ and g_μ for $\mu = 1, 2, \ldots, i-1$.

There is merit in not using some or all constraints (12). First, the system (7) retains its block Vandermonde structure and a fast solver can be deployed [21]. Second, if our assumptions hold, we must have $f_\mu = cf_i(x_1, \ldots, x_\mu, \alpha_{\mu+1}, \ldots, \alpha_i)$ and $g_\mu = cg_i(x_1, \ldots, x_\mu, \alpha_{\mu+1}, \ldots, \alpha_i)$ for some non-zero scalar $c \in \mathsf{K}$. Thus, unlucky anchor points $(\alpha_2, \ldots, \alpha_n)$ may be diagnosed via testing the projections.

In general, there is a trade-off of optimizing the number of evaluations against the arising cost of solving the linear systems. Note that instead of powers $(\xi_1^\ell, \ldots, \xi_i^\ell)$ in (6) one also could use fresh values $(\xi_{1,\ell}, \dots, \xi_{i,\ell})$. The proof of property (10) then uses the idea that for symbolic values $\xi_\mu = v_\mu$ the property is true over the function field $\mathsf{K}(v_1, \dots, v_i)$. □

Remark 2.3. At iteration i we have arbitrarily chosen the variable x_1 for our error-locator polynomial Λ_i. We could also have chosen x_2, or $x_3, \dots$, or x_i. Again, one may select that variable x_μ for which the sets $D_{f,i,E}$ and $D_{g,i,E}$ 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_i is chosen, one gets no overlap in the terms in $D_{f,i,E}$ or $D_{g,i,E}$.

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 Λ_i, one provably needs at most $\sum_{i=1}^{n} t_{f,i-1} t_{g,i-1}(\deg_{x_i}(f) + \deg_{x_i}(g) + 2E + 1) \le ((n-1)t_f t_g + 1)(\max_i\{\deg_{x_i}(f) + \deg_{x_i}(g)\} + 2E + 1)$ values $\beta_{i,\ell}$ (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 δ_j in (4) and η_m in (5) by $\delta_j \le \deg(f_i) - d_{j,1} - \cdots - d_{j,i-1}$, $\eta_m \le \deg(g_i) - e_{m,1} - \cdots - e_{m,i-1}$. This adds additional pairs of degrees $(\deg(f_i), \deg(g_i))$ to Assumption 3. Under Assumption 5, all degrees can be estimated by univariate rational recovery.

First, we show that f_i and g_i are relatively prime with high probability. We consider the substitutions $f_{i,u,x} = f(x_1, x_2 + u_2 x_1, \dots, x_i + u_i x_1, x_{i+1}, \dots, x_n)$ and $g_{i,u,x} = g(x_1, x_2 + u_2 x_1, \dots, x_i + u_i x_1, x_{i+1}, \dots, x_n)$ over the function field $\mathsf{K}_u = \mathsf{K}(u_2, \dots, u_i)$. The map $x_\mu \mapsto x_\mu + u_\mu x_1$, were $2 \le \mu \le i$, constitutes a ring isomorphism on $\mathsf{K}_u[x_1, \dots, x_n]$. Therefore the polynomials $f_{i,u,x}$ and $g_{i,u,x}$ are relatively prime, because their pre-images f and g are relatively prime (extending to K_u cannot change this). Let $\rho_i \in \mathsf{K}_u[x_2, \dots, x_n]$ be the Sylvester resultant of $f_{i,u,x}$ and $g_{i,u,x}$ with respect to the variable x_1. We have $\rho_i \ne 0$ (relative primeness in $\mathsf{K}_u(x_2, \dots, x_n)[x_1]$), and if $\rho(x_2, \dots, x_i, \alpha_{i+1}, \dots, \alpha_n) \ne 0$, the pair $f_{i,u} = f(x_1, x_2 + u_2 x_1, \dots, x_i + u_i x_1, \alpha_{i+1}, \dots, \alpha_n)$ and $g_{i,u} = g(x_1, x_2 + u_2 x_1, \dots, x_i + u_i x_1, \alpha_{i+1}, \dots, \alpha_n)$ is relatively prime in $\mathsf{K}_u(x_2, \dots, x_i)[x_1]$. But the leading coefficients of $f_{i,u}$ and $g_{i,u}$ with respect to x_1 are in K_u, so relative primeness persists in $\mathsf{K}_u[x_1, \dots, x_i]$, and their pre-images f_i and g_i under the inverse isomorphism $x_\mu \mapsto x_\mu - u_\mu x_1$ are relatively prime over K_u, and also over the field K, which contains all coefficients of f_i and g_i.

We finally show how to compute $\deg(f_i)$ and $\deg(g_i)$ by randomization. Cauchy interpolation recovers $f_{i,u}$ and $g_{i,u}$ over $\mathsf{K}_u(x_2, \dots, x_i)$, and also with high probability the images under evaluation $x_\mu \leftarrow \phi_\mu$, $u_\mu \leftarrow \theta_\mu$ for random elements $\phi_\mu, \theta_\mu \in S \subseteq \mathsf{K}$ $(2 \le \mu \le i)$. The evaluations $\phi_2, \dots, \theta_i$ must preserve a non-zero leading subresultant coefficient.

We compute (with high probability) $\deg_{x_1}(f)$ as $\deg(f_1)$ and $\deg_{x_1}(g)$ as $\deg(g_1)$. Similarly, we compute (with high probability) $\deg_{x_i}(f)$ and $\deg_{x_i}(g)$ by rational function recovery (with outlier errors) of $f(\alpha_1, \dots, \alpha_{i-1}, x_i, \alpha_{i+1}, \dots, \alpha_n)/g(\alpha_1, \dots, \alpha_{i-1}, x_i, \alpha_{i+1}, \dots, \alpha_n)$, where $2 \le i \le n$. As a side consequence, we recover (with high probability) the term exponents of x_i in f and g in (1), namely $D_f^{[i]} = \{x_i^{d_{j,i}} \mid 1 \le j \le t_f\}$ and $D_g^{[i]} = \{x_i^{e_{m,i}} \mid 1 \le m \le t_g\}$. Thus, we can further restrict δ_j in (4) and η_m in (5) by $\delta_j \le \deg(f_i) - d_{j,1} - \cdots - d_{j,i-1}, x_i^{\delta_j} \in D_f^{[i]}$, $\eta_m \le \deg(g_i) - e_{m,1} - \cdots - e_{m,i-1}, x_i^{\eta_m} \in D_g^{[i]}$. □

Remark 2.5. Our initialization for $i = 1$ uses for f_1 all terms x_1^δ, where $\delta = 0, 1, \dots, \deg_{x_1}(f)$, and for g_1 all terms x_1^η, where $\eta = 0, 1, \dots, \deg_{x_1}(g)$. By our definitions (4), (5) and (8) we evaluate the fraction at ξ_1^ℓ for $\ell = 0, 1, \dots, L-1$ with $L = \deg_{x_1}(f) + \deg_{x_1}(g) + 1 + 2E$. We suppose that $\xi^{\ell_1} \ne \xi^{\ell_2}$ in the range for ℓ. 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/g. For $L(E) \ge |D_{f,i,E} D_{g,i,E}|$ or, in practice, $L(E) \ge |D_{f,i,E}| + |D_{g,i,E}| + L_0$, which we use in Sections 3 and 4 with $L_0 = 10$, we must attain $k \le L(E)/q \le E$. Note that the latter may for $i \ge 2$ not have a solution for E if the rate $1/q$ is not sufficiently small. We have $|D_{f,i,E}| \le (E+1)|D_{f,i,0}|$ and $|D_{g,i,E}| \le (E+1)|D_{g,i,0}|$ in (4) and (5), so in practice in the worst case $L(E) \le (E+1)|D_{f,i,0}| + (E+1)|D_{g,i,0}| + L_0 \le qE \Longrightarrow q > |D_{f,i,0}| + |D_{g,i,0}|$. □

Remark 2.7. The first algorithm for recovering a sparse rational function without Assumption 5 is described in [19]. As an example, the unreduced 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 $\log(\deg(f))$. Here we have followed the idea of lifting an unreduced fraction by delaying Assumption 5 until after establishing the key Berlekamp/Welch property (10), which is $f_i/g_i = \bar{f}/\bar{g}$. If f_i/g_i is unreduced, the error corrected $\bar{f}$ and $\bar{g}$ may not be equal to the sparse projections f_i and g_i. As we have supposed in [19], the sparsest possibly unreduced fraction f/g of lowest degree can be unique, hence liftable via $\bar{f}$ and $\bar{g}$. The initial sparse f_1 and g_1 can be also obtained by computing a sparse polynomial multiple [10]. Numerically, it may also be possible to recover a sparse unreduced 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 unreduced recovery is also useful when the evaluations at α_μ (see Remark 2.4) do not yield a numerically relatively prime univariate pair f_1, g_1.

In the exact case, there are other ways of determining the coefficients in $\mathsf{K}[x_i]$ 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

recovery (see [16, End of Section 6]). 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_1 and g_1. By using more than one random anchor α_μ, where $2 \leq \mu \leq n$, one can improve the probability that no occurring term is falsely dropped from the term sets for f_i and g_i. □

Remark 2.8. After recovering the $\mathsf{K}[x_i]$ coefficients of f_i and g_i, one may sparsify those coefficients by shifting $x_i = x_i + \sigma_i$, where σ_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, Θ is introduced to measure whether the evaluation is an outlier error, that is, we say the evaluation β at the point $(\zeta_1, \dots, \zeta_n) \in \mathbb{C}^n$ is an outlier error, if $\beta = \gamma + \gamma'$, where $\gamma = f_i(\zeta_1, \dots, \zeta_n)/g_i(\zeta_1, \dots, \zeta_n) \in \mathbb{C} \cup \{\infty\}$, and $|\beta/\gamma| \geq \Theta$. Again false poles and non-poles are allowed; see explanation immediately after Assumption 2. Consider the rational function $f/g \in \mathbb{C}(x_1, \dots, x_n)$, 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_i and g_i when f_{i-1} and g_{i-1} are already computed.

Let $f^{[i]} = f(\alpha_1, \dots, \alpha_{i-1}, x_i, \alpha_{i+1}, \dots, \alpha_n) = \sum_{j=1}^{\bar{d}_f} \psi_j^{[i]} x_i^j$, $g^{[i]} = g(\alpha_1, \dots, \alpha_{i-1}, x_i, \alpha_{i+1}, \dots, \alpha_n) = \sum_{m=1}^{\bar{d}_g} \chi_m^{[i]} x_i^m$, 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 $\bar{d}_f \geq \deg(f)$ and $\bar{d}_g \geq \deg(g)$. 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 $\ell = 0, 1, \dots, \bar{d}_f + \bar{d}_g + 2E + 1$ we compute

$$\beta_\ell = \gamma_\ell + \gamma'_\ell, \text{ where } \gamma_\ell = \frac{f(\zeta^\ell, \alpha_2, \dots, \alpha_n)}{g(\zeta^\ell, \alpha_2, \dots, \alpha_n)} \in \mathbb{C} \cup \{\infty\}, \quad (13)$$

where γ'_ℓ 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 ℓ, such that $|\beta_\ell/\gamma_\ell| \geq \Theta$, is $\leq E$. Having (13), we construct the following linear equations for $\ell = 0, 1, \dots, \bar{d}_f + \bar{d}_g + 2E$,

$$\sum_{j=0}^{\bar{d}_f+E} y_j \zeta^{\ell\, j} - \beta_\ell \sum_{m=0}^{\bar{d}_g+E} z_m \zeta^{\ell\, m} = 0, \quad (14)$$

The above equations form a linear system

$$G \begin{bmatrix} \vec{\boldsymbol{y}} & \vec{\boldsymbol{z}} \end{bmatrix}^T = [V_1, -\Gamma_1 W_1] \begin{bmatrix} \vec{\boldsymbol{y}} & \vec{\boldsymbol{z}} \end{bmatrix}^T = \mathbf{0}, \quad (15)$$

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

$$s = \min(\bar{d}_f - \deg_{x_1}(f), \bar{d}_g - \deg_{x_1}(g)).$$

According to the discussion following Assumption 5 in Section 2, we know that $\rho = 1 + E - k + s$. Having ρ, the linear equations (14) are transformed into the following reduced linear equations by removing some unknown coefficients of higher degree in (14), namely, for $\ell = 0, 1, \dots, \bar{d}_f + \bar{d}_g + 2E$

$$\sum_{j=0}^{\bar{d}_f+E-\rho+1} y_j \zeta^{\ell\, j} - \beta_\ell \sum_{m=0}^{\bar{d}_g+E-\rho+1} z_m \zeta^{\ell\, m} = 0, \quad (16)$$

whose matrix form is

$$\widetilde{G} \begin{bmatrix} \vec{\boldsymbol{y}} & \vec{\boldsymbol{z}} \end{bmatrix}^T = [\widetilde{V}_1, -\Gamma_1 \widetilde{W}_1] \begin{bmatrix} \vec{\boldsymbol{y}} & \vec{\boldsymbol{z}} \end{bmatrix}^T = \mathbf{0}. \quad (17)$$

Note that the numerical rank deficiency of $\widetilde{G}$ is 1, since $\bar{d} + E + 1 - \rho = \bar{d} - s + k$. The coefficient vector $\vec{\boldsymbol{y}}^T$ of $f^{[1]}\Lambda_1$ and the coefficient vector of $g^{[1]}\Lambda_1$ are achieved from the last singular vector of $\widetilde{G}$. Note that Λ_1 should have the form $\Lambda_1 = (x_1 - \zeta_1^{\lambda_1}) \cdots (x_1 - \zeta_1^{\lambda_k})$. In that case, every root ζ^{λ_κ}, $1 \leq \kappa \leq k$, of Λ_1 can be detected by checking for $\ell = 0, 1, \dots, \bar{d}_f + \bar{d}_g + 2E$ with a preset tolerance ϵ_{root}:

$$\ell \in \{\lambda_1, \dots, \lambda_k\} \Longleftrightarrow |(f^{[1]}\Lambda_1)(\zeta^\ell)| + |(g^{[1]}\Lambda_1)(\zeta^\ell)| \leq \epsilon_{\text{root}}.$$

Having Λ_1, we obtain $f^{[1]}$ by applying the approximate univariate polynomial division technique between $f^{[1]}\Lambda_1$ with Λ_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 $D_f^{[i]}$ and $D_g^{[i]}$ of $f^{[i]}$ and $g^{[i]}$.

Remark 3.1. The preset tolerance measures ϵ_{root} and ϵ_{coeff} require that the singular solution vector $[\vec{\boldsymbol{y}}, \vec{\boldsymbol{z}}]^T$ is normalized. We normalize the Euclidean 2-norm to 1. Because we oversample by $\bar{d}_f - \deg_{x_i}(f) + \bar{d}_g - \deg_{x_i}(g)$ evaluations in (16), noisy evaluations can be taken as extra outliers. The justification that $f^{[1]}(\zeta^\ell)$ and/or $g^{[1]}(\zeta^\ell)$ is separated from 0 for (almost) all $\ell \neq \lambda_\kappa$ is from [17, Section 3, Lemma 3.1]. As in [17], we use the same justification for correctly identifying non-zero terms via ϵ_{coeff}, but here an incorrectly dropped term cannot be reintroduced later. Therefore ϵ_{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 ℓ. However, the values in Γ_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 Γ_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

$$\bar{D}_{f,i} = \{x_1^{d_{j,1}} \cdots x_{i-1}^{d_{j,i-1}} x_i^{\delta_j} \mid x_1^{d_{j,1}} \cdots x_{i-1}^{d_{j,i-1}} \in D_{f,i-1}, \quad x_i^{\delta_j} \in D_f^{[i]}, 0 \le \delta_j \le \bar{d}_f - d_{j,1} - \cdots - d_{j,i-1}\}, \tag{18}$$

$$\bar{D}_{g,i} = \{x_1^{e_{m,1}} \cdots x_{i-1}^{e_{m,i-1}} x_i^{\eta_m} \mid x_1^{e_{m,1}} \cdots x_{i-1}^{e_{m,i-1}} \in D_{g,i-1}, \quad x_i^{\eta_m} \in D_g^{[i]}, 0 \le \eta_m \le \bar{d}_g - e_{m,1} - \cdots - e_{m,i-1}\}. \tag{19}$$

Described in Remark 2.4, the new variable x_i is chosen among $x_i, \ldots, x_n$ such that the terms sets $D_{f,i,E}$ in (4) for $f_i\Lambda_i$ and $D_{g,i,E}$ in (5) for $g_i\Lambda_i$ are as small as possible. We designate the possible terms in $f_i\Lambda_i$ and $g_i\Lambda_i$, represented as (4) and (5), as $D_{f,i,E} = \{x_1^{\bar{d}_{j,1}} \cdots x_i^{\bar{d}_{j,i}} \mid j = 1, 2, \ldots, \bar{t}_{f,E}\}$ and $D_{g,i,E} = \{x_1^{\bar{e}_{m,1}} \cdots x_i^{\bar{e}_{m,i}} \mid m = 1, 2, \ldots, \bar{t}_{g,E}\}$. The unknown polynomials $f_i\Lambda_i$ and $g_i\Lambda_i$ are represented as $f_i\Lambda_i = \sum_{j=1}^{\bar{t}_{f,E}} y_j x_1^{\bar{d}_{j,1}} \cdots x_i^{\bar{d}_{j,i}}$, $g_i\Lambda_i = \sum_{m=1}^{\bar{t}_{g,E}} z_m x_1^{\bar{e}_{m,1}} \cdots x_i^{\bar{e}_{m,i}}$, where y_j and z_k are indeterminates.

Let $b_1, \ldots, b_i \in \mathbb{Z}_{>0}$ be sufficient large distinct prime numbers and s_j be random integers with $1 \le s_j < b_j$. We choose $\zeta_j = \exp(2\pi \boldsymbol{i}/b_j)^{s_j} \in \mathbb{C}$ for $1 \le j \le i$ (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 $L \ge \bar{t}_{f,E}\bar{t}_{g,E}$ evaluations. In fact, $\bar{t}_{f,E}\bar{t}_{g,E}$ 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 $L = \bar{t}_{f,E} + \bar{t}_{g,E} + 10$ probes to obtain $f_i\Lambda_i$ and $g_i\Lambda_i$. In the noisy case, we start from the approximate evaluations for $\ell = 0, \ldots, L-1$,

$$\beta_{i,\ell} = \gamma_{i,\ell} + \gamma'_{i,\ell}, \text{ with } \gamma_{i,\ell} = f_i(\zeta_1^\ell, \ldots, \zeta_i^\ell)/g_i(\zeta_1^\ell, \ldots, \zeta_i^\ell), \tag{20}$$

where $\gamma'_{i,\ell}$ is noise or an outlier error. With y_j and z_m unknown, (20) yield the following linear system:

$$G \begin{bmatrix} \vec{\boldsymbol{y}}^T \\ \vec{\boldsymbol{z}}^T \end{bmatrix} = [V_{i,L}(\zeta_1, \ldots, \zeta_i), -\Gamma_{i,L} W_{i,L}(\zeta_1, \ldots, \zeta_i)] \begin{bmatrix} \vec{\boldsymbol{y}}^T \\ \vec{\boldsymbol{z}}^T \end{bmatrix} = \mathbf{0} \tag{21}$$

(cf. (7)), where $L = \bar{t}_{f,E} + \bar{t}_{g,E} + L_0$ with $L_0 \ge 1$ constant, $V_{i,L}, W_{i,L}$ are Vandermonde matrices, and $\Gamma_{i,L} = \operatorname{diag}(\beta_{i,0}, \ldots, \beta_{i,L-1})$. One may estimate the numerical rank deficiency of G, denoted by ρ, by computing its SVD. In consequence, the actual count of errors $k = 1 + E - \rho$ is obtained.

Now let us show how to compute the coefficients of $f_i\Lambda_i$ and $g_i\Lambda_i$. Having the actual count of errors k, the possible terms in $f_i\Lambda_i$ and the possible terms in $g_i\Lambda_i$ are represented precisely by $D_{f,i,k}$ and $D_{g,i,k}$ instead of $D_{f,i,E}, D_{g,i,E}$. Furthermore, the numerical rank deficiency of $\widetilde{G}$, produced by (20) with fewer terms, is 1. In the sequel, the coefficient vector of $f_i\Lambda_i$ and $g_i\Lambda_i$ is achieved from the last singular vector of $\widetilde{G}$. Because all roots of Λ_i have the form of $\zeta_1^{\lambda_\kappa}$, $1 \le \kappa \le k$, similarly to the univariate case all roots can be identified by checking the evaluations $\ell \in \{\lambda_1, \ldots, \lambda_k\}$

$$\iff |(f_i\Lambda_i)(\zeta_1^\ell, \ldots, \zeta_i^\ell)| + |(g_i\Lambda_i)(\zeta_1^\ell, \ldots, \zeta_i^\ell)| \le \epsilon_{\text{root}}. \tag{22}$$

The remaining task is to compute f_i and g_i from the three polynomials $\Lambda_i, f_i\Lambda_i, g_i\Lambda_i$, which constitutes an approximate polynomial division problem. Since $\Lambda_i(x_1)$ is a univariate polynomial, Λ_i is the content of $f_i\Lambda_i$ and $g_i\Lambda_i$ w.r.t. the variables $x_2, \ldots, x_i$, and the corresponding primitive parts are f_i and g_i, respectively (note that f_i and g_i are assumed to be relatively prime—see Remark 2.4). Thus approximate univariate polynomial division can be employed to compute f_i and g_i. We then further get the exact supports of f_i and g_i by removing terms whose coefficients are in absolute value $\le \epsilon_{\text{coeff}}$. Remark 3.1 is relevant again, now with oversampling by $L_0 = 10$.

Alternatively, one may compute f_i, g_i from Λ_i via the error locations. In fact, λ_κ is the location of an outlier error if $\zeta_i^{\lambda_\kappa}$ is a root of Λ_i. In this case, all error locations $\lambda_1, \ldots, \ldots \lambda_k$ can be determined by (22). By removing all the evaluations at $\lambda_1, \ldots, \lambda_k$, one gets the approximate evaluations without outlier errors, that is, for $\ell = 0, \ldots, L-1$, $\ell \notin \{\lambda_1, \ldots, \lambda_k\}$,

$$\beta_{i,\ell} \approx f_i(\zeta_1^\ell, \ldots, \zeta_i^\ell)/g_i(\zeta_1^\ell, \ldots, \zeta_i^\ell). \tag{23}$$

In this situation, the remaining problem can be transformed as the problem of interpolating f_i, g_i from their possible terms $\bar{D}_{f,i}$ and $\bar{D}_{g,i}$ and the approximate evaluations (23). 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: ▸ $\frac{f(x_1,\ldots,x_n)}{g(x_1,\ldots,x_n)} \in \mathbb{C}(x_1, \ldots, x_n)$ input as a black box with noise and outlier errors, the latter at a given rate (see Remark 1.1).

▸ $(x_1, \ldots, x_n)$: an ordered list of variables in f/g.

▸ $\bar{d}, \bar{e}$: total degree bounds $\bar{d} \ge \deg(f)$ and $\bar{e} \ge \deg(g)$.

▸ $\epsilon_{\text{coeff}} > 0$ (for "forcing underflow" of terms), $\epsilon_{\text{root}} > 0$ (for zero detection), the given tolerance.

Output: $f(x_1, \ldots, x_n)/c$ and $g(x_1, \ldots, x_n)/c$, where $c \in \mathbb{C}$.

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 $f^{[i]}$ and $g^{[i]}$ and get their supports $D_f^{[i]}$ and $D_g^{[i]}$:

(a) Choose a random root of unity ζ and get the evaluations $\beta_{i,\ell}$ with the noise and the outlier errors as (13).

(b) Construct the matrix G in (15) from $\beta_{i,\ell}$ and ζ. Compute the SVD of G and find its numerical rank deficiency r. A relative tolerance ϵ_{rank} for a jump in the singular values can be provided as an additional input.

(c) Get the matrix $\widetilde{G}$ from the reduced linear system (16) with r, and then obtain $f^{[i]}\Lambda_i$ and $g^{[i]}\Lambda_i$ from the last singular vector of $\widetilde{G}$.

(d) Get the error locator polynomial Λ_i by checking (22).

(e) Obtain $f^{[i]}$ and $g^{[i]}$ by applying univariate polynomial division, and then get the actual support $D_f^{[i]}$ of $f^{[i]}$ by rounding coefficients that are absolutely $\le \epsilon_{\text{coeff}}$ to 0. Similarly, get the actual support $D_g^{[i]}$ of $g^{[i]}$.

3. Let $D_{f,1} = D_f^{[1]}$ and $D_{g,1} = D_g^{[1]}$. For $i = 2, \ldots, n$ do:
   Interpolate the polynomials f_i and g_i as follows:

(a) Choose the variable x_μ from $x_1, \ldots, x_i$ such that $D_{f,i,E}$ and $D_{g,i,E}$ have the fewest elments.

(b) Compute $f_i\Lambda_i$ and $g_i\Lambda_i$:

(b.1) Choose random roots of unity $\zeta_1, \ldots, \zeta_i$. For $\ell = 0, 1, 2, \ldots$, compute the evaluations $\beta_{i,\ell}$ with the noise and the outlier errors as (20).

(b.2) Construct the matrix G in (21) from $\beta_{i,\ell}$ 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.

(b.4) Reconstruct the possible terms $D_{f,i,k}, D_{g,i,k}$ in $f_i\Lambda_i$ and $g_i\Lambda_i$.

(b.5) Get the shrunk matrix $\widetilde{G}$ from $\beta_{i,\ell}$ and $D_{f,i,k}, D_{f,i,k}$.

(b.6) Obtain $f_i\Lambda_i$, $g_i\Lambda_i$ from the last singular vector of $\widetilde{G}$.

(c) Obtain f_i and g_i

(c.1) Find the error location $\lambda_1, \ldots, \lambda_k$, for which $\zeta_\mu^{\lambda_\kappa}$ is the root of Λ_i.

(c.2) Get the approximate evaluations without outlier errors by removing ones correspond to λ_κ.

(c.3) Interpolate 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 $\beta_{n,\ell}$ as (23) and the exact terms $D_{f,n}$ and $D_{f,n}$.

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

(c) Obtain $f(x_1, \ldots, x_n)/c$ and $g(x_1, \ldots, x_n)/c$ from $\vec{\boldsymbol{y}}, \vec{\boldsymbol{z}}$ 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 *Digits*:=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 $-5 \le c \le 5$. Here *Random Noise* denotes the range of relative noise randomly added to the black box evaluations of f/g; $\bar{d}_f \ge \deg(f)$ and $\bar{g}_g \ge \deg(g)$ 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 $(\|c\tilde{f} - f\|_2^2 + \|c\tilde{g} - g\|_2^2)/(\|f\|_2^2 + \|g\|_2^2)$, where $\tilde{f}/\tilde{g}$ 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 $0.01 \times [100, 200]$ or $0.01 \times [200, 300]$. 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.

Ex.	*Random Noise*	$\bar{d}_f, \bar{d}_g$	deg(f), deg(g)	t_f, t_g	$1/q$	E	*Time (secs.)*	*Rel. Error*
1	$10^{-4} \sim 10^{-2}$	10, 10	3, 3	1, 3	1/3	37	5.9	6.0e–7
2	$10^{-5} \sim 10^{-3}$	6, 6	4, 5	2, 4	1/3	39	4.4	3.1e–6
3	$10^{-6} \sim 10^{-4}$	18, 13	8, 3	4, 3	1/4	26	1.5	2.3e–8
4	$10^{-5} \sim 10^{-3}$	20, 20	10, 10	4, 4	1/3	52	8.4	2.6e–4
5	$10^{-6} \sim 10^{-4}$	18, 30	3, 15	2, 6	1/4	30	9.9	8.8e–8
6	$10^{-6} \sim 10^{-4}$	40, 40	20, 20	5, 5	1/4	48	32	6.2e–9
7	$10^{-6} \sim 10^{-4}$	50, 30	30, 7	6, 3	1/5	34	24	8.6e–6
8	$10^{-7} \sim 10^{-5}$	30, 70	5, 40	4, 7	1/4	61	57	2.7e–9
9	$10^{-7} \sim 10^{-5}$	80, 80	50, 50	5, 5	1/5	52	29	7.2e–10
10	$10^{-7} \sim 10^{-5}$	80, 80	50, 50	51, 51	1/8	31	23	3.2e–7

Table 1: Performance: univariate case

Remark 4.1. In our tests, the k outlier errors are introduced in random locations after the bound $E \ge k$ is derived from the error rate $1/q$. However, our algorithm also makes random choices, the anchor points α and the random roots of unity ζ (see Step 2(a)). We perform 20 trials of the ζ's before giving up with recovery. Running times can fluctuate as a new random choice for ζ 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 $1/q = 1/4$. □

Ex	*Random Noise*	$\bar{d}_f, \bar{d}_g$	deg(f), deg(g)	t_f, t_g	n	$1/q$	E	N	*time secs.*	*Rel. Error*
1	10^{-5}∼10^{-3}	3, 3	1, 1	2, 2	2	1/10	12	403	6.2	7.3e–7
2	10^{-5}∼10^{-3}	5, 5	2, 2	3, 3	2	1/12	21	306	6.0	4.5e–8
3	10^{-5}∼10^{-3}	2, 5	1, 4	2, 4	3	1/15	13	561	13	4.7e–7
4	10^{-6}∼10^{-4}	8, 8	5, 2	10, 6	3	1/40	12	616	47	3.6e–6
5	10^{-7}∼10^{-5}	10, 10	7, 7	10,10	5	1/90	7	1508	197	5.1e–11
6	10^{-7}∼10^{-5}	15, 10	10, 3	15, 5	8	1/90	7	2423	273	7.4e–11
7	10^{-7}∼10^{-5}	10, 15	5, 13	4, 6	10	1/80	2	1289	24	8.1e–10
8	10^{-7}∼10^{-5}	25, 25	20, 20	7, 7	15	1/100	3	2890	137	2.9e–10
9	10^{-8}∼10^{-6}	35, 35	30, 30	6, 6	20	1/80	2	3881	230	5.5e–13
10	10^{-8}∼10^{-6}	45, 45	40, 40	6, 6	5	1/80	6	2080	219	3.7e–12
11	10^{-8}∼10^{-6}	85, 85	60, 60	7, 7	4	1/100	11	2787	1479	3.7e–13
12	10^{-8}∼10^{-6}	85, 85	80, 80	3, 3	5	1/30	4	1773	83	4.5e–12
13	10^{-9}∼10^{-7}	70, 0	40, 0	6, 1	15	1/70	2	2284	75	7.5e–18
14	10^{-8}∼10^{-6}	25, 25	20, 20	5, 5	102	1/80	1	10191	272	6.1e–12

Table 2: Performance: multivariate case

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 $-5 \le c \le 5$. Here *Random Noise* denotes the noise in this range randomly added to the black box of f/g; $\bar{d}_f \ge \deg(f)$ and $\bar{g}_g \ge \deg(g)$ 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 $\bar{d}_f$ and $\bar{d}_g$, consumes $(\bar{d}_f + \bar{d}_g + 2E + L_0) \cdot 102$ probes, about 5000. We then use x_i as the variable in the error locator polynomial Λ_i; see Remark 2.3. We have for each i the estimates $|D_{f,i,E}| = t_f(\deg_{x_i}(f) + 1 + E) = 5(3 + 1 + 1)$ and $|D_{g,i,E}| = t_g(\deg_{x_i}(g) + 1 + E) = 5(2 + 1 + 1)$, that is $45 + L_0$ evaluations, or about 5000 in total. Using sharper upper bounds for $\deg_{x_i}(f)$ and $\deg_{x_i}(g)$ 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 $d_{j,\mu}$, $e_{m,\mu}$ to be determined.

Ex	Random Noise	Rel. Outlier Error Θ	deg(f), deg(g)	t_f, t_g	n	1/q	E	N	time secs	Rel. Error
1	$10^{-4}\sim10^{-2}$	$1\sim2$	5, 5	2, 3	1	1/4	24	94	0.8	8.1e–4
2	$10^{-5}\sim10^{-3}$	$0.1\sim0.2$	15, 15	3, 5	1	1/10	9	84	1.5	3.3e–4
3	$10^{-7}\sim10^{-5}$	$0.001\sim0.002$	20, 10	4, 3	1	1/15	5	74	0.6	9.3e–10
4	$10^{-6}\sim10^{-4}$	$0.01\sim0.02$	30, 25	4, 4	1	1/7	12	84	0.8	9.4e–9
5	$10^{-7}\sim10^{-5}$	$0.001\sim0.002$	50, 40	5, 4	1	1/40	3	288	3.0	8.1e–4
6	$10^{-5}\sim10^{-3}$	$0.1\sim0.2$	5, 8	1, 3	2	1/30	2	200	1.9	1.8e–2
7	$10^{-6}\sim10^{-4}$	$0.01\sim0.02$	10, 15	3, 3	4	1/40	2	860	8.2	9.5e–9
8	$10^{-7}\sim10^{-5}$	$0.01\sim0.02$	10, 10	3, 2	15	1/40	2	2433	18	3.0e–12
9	$10^{-8}\sim10^{-6}$	$0.01\sim0.02$	8, 8	4, 3	30	1/50	2	3236	35	1.2e–12
10	$10^{-9}\sim10^{-7}$	$0.01\sim0.02$	15, 15	3, 3	50	1/60	1	5299	50	2.5e–10

Table 3: Performance: small outliers

In Table 3 we give first tests with small outlier errors; see Remark 3.1. *Outlier Error* denotes the relative outlier error Θ, which is randomly selected in the given range.

Example 4.1. We now demonstrate the Candes-Tao recovery of unreduced sparse rational functions discussed in Remark 2.7, that on a small example with no outlier errors ($k = E = 0$): let $f = (x^{11}+1)(x-1) = x^{12} - x^{11} + x - 1$, $\bar{d}_f = 12$, and $g = (x+1)(x^5-1)$, $\bar{d}_g = 6$, $L = \bar{d}_f + \bar{d}_g + 1 = 19$. We have $f/g =$

$$\frac{x^{12}-x^{11}+x-1}{x^6+x^5-x-1} = \frac{x^{10}-x^9+x^8-x^7+x^6-x^5+x^4-x^3+x^2-x+1}{x^4+x^3+x^2+x+1}. \quad (24)$$

We compute $\beta_\ell = \gamma_\ell = (f/g)(\zeta^{\ell+1})$ for $\zeta = \exp(2\pi\boldsymbol{i}/31)$ and $\ell = 0, 1, \ldots, L-1$ in hardware precision complex floating point arithmetic. The shift in the exponent to $\ell + 1$ avoids $g(1) = 0$. Now we solve the linear system (15) for a real vector $[\vec{\boldsymbol{y}}, \vec{\boldsymbol{z}}]^T \in \mathbb{R}^{19}$ with the following constraint: $y_{12} = 1$, 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

$$\begin{bmatrix} \text{Realpart}(V_1), -\text{Realpart}(\Gamma_1 W_1) \\ \text{Imagpart}(V_1), -\text{Imagpart}(\Gamma_1 W_1) \end{bmatrix} \begin{bmatrix} \vec{\boldsymbol{y}}^T \\ \vec{\boldsymbol{z}}^T \end{bmatrix} = \mathbf{0}, y_{12} = 1. \quad (25)$$

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 $\sum_j |y_j| + \sum_m |z_m| = \|[\vec{\boldsymbol{y}}, \vec{\boldsymbol{z}}]\|_1$ via Tshebysheff's linear programming formulation. In our case, Maple 16's call to `Optimization['LPSolve']` with `method = activeset` produces the solution $f = x^{12} - 1.0\,x^{11} - 1.78\times10^{-10}\,x^{10} + 1.72\times10^{-10}\,x^9 + 1.72\times10^{-10}\,x^7 - 1.78\times10^{-10}\,x^6 + 1.82\times10^{-10}\,x^5 - 1.88\times10^{-10}\,x^4 + 1.92\times10^{-10}\,x^3 - 1.88\times10^{-10}\,x^2 + 1.0\,x - 1.0$ and $g = 1.0\,x^6 + 1.0\,x^5 - 5.83\times10^{-12}\,x^3 - 2.07\times10^{-12}\,x^2 - 1.0\,x - 1.0$ with an objective value of 6.99999999999878. 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

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

[2] Boyer, B., Comer, M. T., and Kaltofen, E. L. Sparse polynomial interpolation by variable shift in the presence of noise and outliers in the evaluations. In *Electr. Proc. Tenth Asian Symposium on Computer Mathematics (ASCM 2012)* (2012). .

[3] Candes, E., and Tao, T. Decoding by linear programming. *IEEE Trans. Inf. Theory* IT-*51*, 12 (2005), 4203–4215.

[4] Comer, M. T., Kaltofen, E. L., and Pernet, C. Sparse polynomial interpolation and Berlekamp/Massey algorithms that correct outlier errors in input values. In *ISSAC 2012 Proc. 37th Internat. Symp. Symb. Alg. Comput.* (New York, N. Y., July 2012), J. van der Hoeven and M. van Hoeij, Eds., ACM, pp. 138–145. .

[5] Cuyt, A., and Lee, W. Sparse interpolation of multivariate rational functions. *Theoretical Comput. Sci. 412* (2011), 1445–1456.

[6] de Kleine, J., Monagan, M., and Wittkopf, A. Algorithms for the non-monic case of the sparse modular GCD algorithm. In *ISSAC'05 Proc. 2005 Internat. Symp. Symb. Alg. Comput.* (New York, N. Y., 2005), M. Kauers, Ed., ACM Press, pp. 124–131.

[7] Giesbrecht, M., Kaltofen, E., and Lee, W. Algorithms for computing sparsest shifts of polynomials in power, Chebychev, and Pochhammer bases. *J. Symbolic Comput. 36*, 3–4 (2003), 401–424. .

[8] Giesbrecht, M., Labahn, G., and Lee, W. Symbolic-numeric sparse interpolation of multivariate polynomials (Ext. Abstract). In *Proc. 9th Rhine Workshop Comput. Alg. (RWCA'04), Univ. Nijmegen, the Netherlands* (2004), pp. 127–139.

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

[10] Giesbrecht, M., Roche, D. S., and Tilak, H. Computing sparse multiples of polynomials. In *Proc. Internat. Symp. on Algorithms and Computation (ISAAC 2010)* (2010).

[11] Kaltofen, E. Greatest common divisors of polynomials given by straight-line programs. *J. ACM 35*, 1 (1988), 231–264. .

[12] Kaltofen, E. Factorization of polynomials given by straight-line programs. In *Randomness and Computation*, S. Micali, Ed., vol. 5 of *Advances in Computing Research*. JAI Press Inc., Greenwhich, Connecticut, 1989, pp. 375–412. .

[13] Kaltofen, E., and Koiran, P. Expressing a fraction of two determinants as a determinant. In *ISSAC 2008* (New York, N. Y., 2008), D. Jeffrey, Ed., ACM Press, pp. 141–146. .

[14] Kaltofen, E., and Pernet, C. Cauchy interpolation with errors in the values. Manuscript in preparation, Jan. 2013.

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

[16] Kaltofen, E., and Yang, Z. On exact and approximate interpolation of sparse rational functions. In *ISSAC 2007 Proc. 2007 Internat. Symp. Symb. Alg. Comput.* (New York, N. Y., 2007), C. W. Brown, Ed., ACM Press, pp. 203–210. .

[17] Kaltofen, E., Yang, Z., and Zhi, L. On probabilistic analysis of randomization in hybrid symbolic-numeric algorithms. In *SNC'07 Proc. 2007 Internat. Workshop Symb.-Numer. Comput.* (New York, N. Y., 2007), J. Verschelde and S. M. Watt, Eds., ACM Press, pp. 11–17. .

[18] Kaltofen, E. L., Lee, W., and Yang, Z. Fast estimates of Hankel matrix condition numbers and numeric sparse interpolation. In *SNC'11 Proc. 2011 Internat. Workshop Symb.-Numer. Comput.* (New York, N. Y., June 2011), M. Moreno Maza, Ed., ACM, pp. 130–136. .

[19] Kaltofen, E. L., and Nehring, M. Supersparse black box rational function interpolation. In *Proc. 2011 Internat. Symp. Symb. Alg. Comput. ISSAC 2011* (New York, N. Y., June 2011), A. Leykin, Ed., ACM, pp. 177–185. .

[20] Olesh, Z., and Storjohann, A. The vector rational function reconstruction problems. In *Proc. Waterloo Workshop on Computer Algebra: devoted to the 60th birthday of Sergei Abramov (WWCA)* (2007), pp. 137–149.

[21] Olshevsky, V., and Shokrollahi, M. A. A displacement approach to decoding algebraic codes. In *Algorithms for Structured Matrices: Theory and Applications*. AMS, Providence, RI, 2003, pp. 265–292. Contemp. Math., vol. 323.

[22] Saraf, S., and Yekhanin, S. Noisy interpolation of sparse polynomials, and applications. In *Proc. 26th Annual IEEE Conf. Comp. Complexity* (2011), IEEE Comp. Soc., pp. 86–92.

[23] Welch, L. R., and Berlekamp, E. R. Error correction of algebraic block codes. US Patent 4,633,470, 1986. Filed 1983; see http://patft.uspto.gov/.

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