On Exact and Approximate Interpolation of Sparse Rational Functions*

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

Zhengfeng Yang
Department of Mathematics
North Carolina State University
Raleigh, North Carolina 27695-8205, USA
zyang4@math.ncsu.edu

ABSTRACT

The black box algorithm for separating the numerator from the denominator of a multivariate rational function can be combined with sparse multivariate polynomial interpolation algorithms to interpolate a sparse rational function. Randomization and early termination strategies are exploited to minimize the number of black box evaluations. In addition, rational number coefficients are recovered from modular images by rational vector recovery. The need for separate numerator and denominator size bounds is avoided via self-correction, and the modulus is minimized by use of lattice basis reduction, a process that can be applied to sparse rational function vector recovery itself. Finally, one can deploy the sparse rational function interpolation algorithm in the hybrid symbolic-numeric setting when the black box for the rational function returns real and complex values with noise. We present and analyze five new algorithms for the above problems and demonstrate their effectiveness on a benchmark implementation.

Categories and Subject Descriptors: I.2.1 [Symbolic and Algebraic Manipulation]: Algorithms

General Terms: algorithms, experimentation

Keywords: sparse rational function interpolation, early termination, hybrid symbolic-numeric computation, rational vector recovery, lattice basis reduction

1. INTRODUCTION

In [16] Kaltofen and Trager present a general method for evaluating separately the numerator and denominator of a rational function in n variables given by "black box" procedure that evaluates the rational function at a point (see Figure 1).

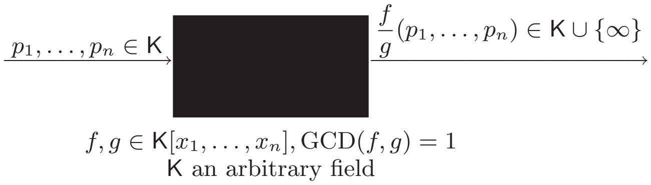
$p_1, \ldots, p_n \in \mathsf{K}$
$\frac{f}{g}(p_1, \ldots, p_n) \in \mathsf{K} \cup \{\infty\}$
$f, g \in \mathsf{K}[x_1, \ldots, x_n], \mathrm{GCD}(f, g) = 1$
K an arbitrary field

Figure 1: Black box for rational function evaluation

It is assumed that the black box procedure returns ∞ if $g(p_1, \ldots, p_n) = 0$. The separation algorithm computes the values of $f(p_1, \ldots, p_n)/c \in \mathsf{K}$ and $g(p_1, \ldots, p_n)/c \in \mathsf{K}$ for $p_1, \ldots, p_n$ in the coefficient field K, where $c \in \mathsf{K} \setminus \{0\}$ is a fixed constant that selects the same associates of the numerator and denominator polynomials for all evaluations. It is observed in [16] that the evaluation procedure can be combined with any sparse polynomial interpolation algorithm to compute the sparse representations of f and g, namely

$$f/c = \sum_{j=1}^{t_f} \psi_j x_1^{d_{j,1}} \cdots x_n^{d_{j,n}}, \; g/c = \sum_{k=1}^{t_g} \chi_k x_1^{e_{k,1}} \cdots x_n^{e_{k,n}},$$

where $\psi_j, \chi_k \in \mathsf{K} \setminus \{0\}$. Here we consider the sparse polynomial interpolation algorithm in [13], which minimizes the number of polynomial evaluations using the early termination paradigm. Our goal is to minimize the number of rational function evaluations in practice. Other work on sparse rational function interpolation is [8], which focuses on general decidability and complexity questions.

The combination of both the separation and the early termination algorithms allows two major speedups. First, in [16] a homotopy is used so that the value of $f/c(p_1, \ldots, p_n)$ can be computed even when $g(p_1, \ldots, p_n) = 0$. However, the algorithm in [13] can be performed for points $p_1^\tau, \ldots, p_{i-1}^\tau, p_i^\kappa, p_{i+1}, \ldots, p_n$ where p_i $(1 \leq i \leq n)$ that are uniformly randomly selected from a sufficiently larger finite subset $S \subseteq \mathsf{K}$. We present the probabilistic analysis for a separation procedure without a homotopy for such random points (see Lemma 2.2). The change avoids the interpolation of a second variable. Second, the sparse interpolation algorithm of [13] is executed simultaneously on f/c and g/c. Thus the early termination pruning techniques can be extended for obtaining numerator and denominator values: if a (partial) numerator or denominator polynomial is known to be complete, further polynomial values can be computed directly without rational recovery (see Section 3).

In the spirit of the early termination paradigm, we also improve our technique for determining the numerator degree $\deg(f)$ and the denominator degree $\deg(g)$ (see Case $\bar{d} \geq$

[*] This material is based on work supported in part by the National Science Foundation under Grants Nos. CCR-0305314 and CCF-0514585.

ISSAC'07, July 29–August 1, 2007, Waterloo, Ontario, Canada.
Copyright 2007 ACM 978-1-59593-743-8/07/0007 ...$5.00.

$\deg(f)$ and $\bar{e} \ge \deg(g)$ on page 206). We have implemented our algorithm and demonstrate the performance on a selection of sparse rational functions (see Section 5). An additional place for improvement arises when recovering numerator and denominator polynomials with integer coefficients, i.e., the case $\mathsf{K} = \mathbb{Q}$. The algorithm in [13] uses modular arithmetic ($\mathsf{K} = \mathbb{Z}_M$, where M is prime) and rational number recovery [27, 15]. It is possible to probabilistically recover the common denominator of the rational coefficients without individual numerator and denominator bounds [21], but then M needs to be larger than is necessary with accurate bounds. Here we take advantage of the fact that several rational numbers are recovered simultaneously and employ the algorithm by [15] in a self-correcting manner, again without any individual numerator and denominator size bounds. Furthermore, we have implemented a rational vector recovery procedure based on Largarias's [20] good simultaneous diophantine approximation algorithm. Via lattice basis reduction, we can for certain inputs further reduce the size of the modulus M. We describe our two algorithms in Section 4.

Finally, we investigate how numerical sparse interpolation algorithms [7] can be used together with our approach on numerical rational function black boxes, i.e., procedures that return the value of the rational function at a point as a floating point number that is an approximation of the exact value (contains "noise"). By making necessary changes in the procedure for separately evaluating the numerator and denominator, we are able to recover low degree sparse numerator and sparse denominator polynomials from approximate values. We describe our approximate algorithm and some remaining issues in Section 6. Our approximate algorithm is related to numeric multivariate rational interpolation (see, e.g., [1]). We note that our methods do not fit a set of given data points, which in the multivariate case leads to multiple solutions, but recovers a sparse rational function uniquely by evaluating at certain points.

2. EVALUATION OF THE NUMERATOR AND DENOMINATOR

We first present an algorithm that returns the values at certain random points of fixed associates of the numerator and denominator polynomial for the black box of the rational function f/g. The algorithm makes use of a univariate rational function recovery procedure, which we summarize for later reference in the following lemma [16, Lemma 1 on p. 315]

LEMMA 2.1. *Let $\bar{d}$ and $\bar{e}$ be non-negative integers, and let $F(X), G(X), H(X) \in \mathsf{K}[X]$, K an arbitrary field, $\deg(H) < \bar{d}+\bar{e}+1$, $\gcd(F,G) = 1$; furthermore, let i_k, $1 \le k \le \bar{d}+\bar{e}+1$, be not necessarily distinct elements in K such that*

$$F \equiv G\,H \pmod{(X-i_1)\cdots(X-i_{\bar{d}+\bar{e}+1})}.$$

Define $h_0(X) := (X-i_1)\cdots(X-i_{\bar{d}+\bar{e}+1})$, $\delta_0 := \bar{d}+\bar{e}+1$, and $h_1(X) := H(X)$, $\delta_1 := \deg(H)$. Now let $h_l(X), q_l(X) \in \mathsf{K}[X]$ be the l-th remainders and quotients respectively, in the Euclidean polynomial remainder sequence

$$h_{l-2}(X) = q_l(X)h_{l-1}(X) + h_l(X),\ \delta_l := \deg(h_l) < \delta_{l-1},\ l \ge 2.$$

In the exceptional case $H = 0$ the sequence is defined to be empty. Finally, let $w_l(X), g_l(X) \in \mathsf{K}[X]$ be the multipliers in the extended Euclidean scheme $w_l h_0 + g_l h_1 = h_l$, namely,

$$w_0 := g_1 := 1, \quad w_1 := g_0 := 0,$$
$$w_l := w_{l-2} - q_l w_{l-1}, \quad g_l := g_{l-2} - q_l g_{l-1} \quad \text{for } l \ge 2.$$

Then there exists an index j, $1 \le j$, such that $\delta_j \le \bar{d} < \delta_{j-1}$. For that index we have

$$\left.\begin{aligned} h_j \equiv g_j H \quad &\pmod{(X-i_1)\cdots(X-i_{\bar{d}+\bar{e}+1})} \\ &\text{and} \quad \deg(g_j) \le \bar{e}. \end{aligned}\right\} \tag{1}$$

Furthermore, if $\bar{d} \ge \deg(F)$ and $\bar{e} \ge \deg(G)$ then $F = ch_j$, $G = cg_j$ for some $c \in \mathsf{K}$.

Our idea is similar to the one in [10, 16]. We obtain the values $f(p_1,\dots,p_n)/c \in \mathsf{K}$ and $g(p_1,\dots,p_n)/c \in \mathsf{K}$ at $p_1,\dots,p_n \in \mathsf{K}$ by selecting once and for all random shift values $B_2,\dots,B_n \in \mathsf{K}$ and by performing univariate rational function recovery for

$$\frac{f(X, B_2X - B_2p_1 + p_2, \dots, B_nX - B_np_1 + p_n)}{g(X, B_2X - B_2p_1 + p_2, \dots, B_nX - B_np_1 + p_n)}. \tag{2}$$

Here the shift values B_i with high probability guarantee that the leading coefficient of the denominator $g(X, B_2X - B_2p_1 + p_2, \dots, B_nX - B_np_1 + p_n)$, say, is independent of the p_i. By making that leading coefficient monic one then can select the same associates for any point $p_1,\dots,p_n$. The values are computed by the evaluation $X = p_1$. Aside from our condition for the B_i, we also need to guarantee that the fraction (2) cannot be reduced by a univariate polynomial GCD (and hence the denominator does not evaluate to 0). That we can enforce probabilistically by choosing the points $p_1,\dots,p_n$ randomly. The sparse polynomial interpolation algorithm in [13], which we will deploy in Section 3, requires the polynomial values at $p_1^{\tau},\dots,p_{i-1}^{\tau},p_i^{\kappa},p_{i+1},\dots,p_n$ as well. Our next lemma shows that those also remain usable with high probability.

LEMMA 2.2. *Let $f,g \in \mathsf{K}[x_1,\dots,x_n]$ with $\mathrm{GCD}(f,g) = 1$, let $d = \deg(f)$ and $e = deg(g)$ and let $t \ge 1$. Furthermore, let $B_2,\dots,B_n \in \mathsf{K}$ be such that $\lambda_1(B_2,\dots,B_n) \ne 0$ where $\lambda_1(\beta_2,\dots,\beta_n)$ is the leading coefficient in X of $g(X, \beta_2 X, \dots, \beta_n X) \in (\mathsf{K}[\beta_2,\dots,\beta_n])[X]$. Finally, for $J \ge 1$ and $t \ge 1$ let*

$$\{(\tau_{j,1},\dots,\tau_{j,n}) \mid 1 \le \tau_{j,k} \le t \textit{ for all } 1 \le j \le J, 1 \le k \le n\},$$
$$(\tau_{1,1},\dots,\tau_{1,n}) = (1,\dots,1)$$

be a set of J distinct exponent vectors. Suppose $p_1,\dots,p_n \in S$ are chosen randomly and uniformly from a finite set $S \subseteq \mathsf{K}$ of cardinality $|S|$. In addition, for $j \ge 1$ let

$$\left.\begin{aligned} f_{1,j}(X) = f(X, B_2X - B_2p_1^{\tau_{j,1}} + p_2^{\tau_{j,2}}, \dots, \\ B_nX - B_np_1^{\tau_{j,1}} + p_n^{\tau_{j,n}}), \\ g_{1,j}(X) = g(X, B_2X - B_2p_1^{\tau_{j,1}} + p_2^{\tau_{j,2}}, \dots, \\ B_nX - B_np_1^{\tau_{j,1}} + p_n^{\tau_{j,n}}). \end{aligned}\right\} \tag{3}$$

Then we have the following probability estimate:

$$\begin{aligned} \mathit{Prob}(\mathrm{GCD}(f_{1,j}(X), g_{1,j}(X)) = 1 \textit{ for all } 1 \le j \le J) \\ \ge 1 - \frac{2((J-1)t+1)\deg(f)\deg(g)}{|S|} \end{aligned}$$

Proof: We first settle the case $J = 1$. For new variables $X, \alpha_1,\dots,\alpha_n$ we define the map:

$$\phi_1 \colon K[x_1, x_2, \dots, x_n, \alpha_1] \to K[X, \alpha_1, \dots, \alpha_n]$$

where

$$\begin{aligned} x_1 &\mapsto X, \\ x_i &\mapsto B_i(X-\alpha_1)+\alpha_i \quad \text{for all} \quad 2 \le i \le n, \\ \alpha_1 &\mapsto \alpha_1. \end{aligned}$$

Namely,

$$\begin{aligned} &\phi_1(h(x_1, x_2, \ldots, x_n, \alpha_1)) \\ &\quad = h(X, B_2(X-\alpha_1)+\alpha_2, \ldots, B_n(X-\alpha_1)+\alpha_n, \alpha_1). \end{aligned}$$

The map ϕ_1 is a ring isomorphism by virtue of the inverse map

$$\begin{aligned} \phi_1^{-1}(X) &= x_1, \\ \phi_1^{-1}(\alpha_1) &= \alpha_1, \\ \phi_1^{-1}(\alpha_i) &= x_i - B_i(x_1-\alpha_1) \quad \text{for all} \quad 2 \le i \le n. \end{aligned}$$

Namely,

$$\begin{aligned} &\phi_1^{-1}(h(X, \alpha_1, \ldots, \alpha_n)) \\ &\quad = h(x_1, \alpha_1, x_2 - B_2(x_1-\alpha_1), \ldots, x_n - B_n(x_1-\alpha_1)). \end{aligned}$$

Next, we prove that $\mathrm{GCD}(\phi_1(f), \phi_1(g)) = 1$. Suppose $\mathrm{GCD}(\phi_1(f), \phi_1(g)) = \hat{h}_1$. Then we have $\phi_1(f) = \hat{f}_1\hat{h}_1$, $\phi_1(g) = \hat{g}_1\hat{h}_1$, for $\hat{f}_1$, $\hat{g}_1$, $\hat{h}_1 \in \mathsf{K}[X, \alpha_1, \ldots, \alpha_n]$. We know that $f = \phi_1^{-1}(\hat{f}_1)\phi_1^{-1}(\hat{h}_1)$ and $g = \phi_1^{-1}(\hat{g}_1)\phi_1^{-1}(\hat{h}_1)$. Now the variable α_1 vanishes in the polynomials f and g. Therefore, α_1 also vanishes in the polynomials $\phi_1^{-1}(\hat{f}_1), \phi_1^{-1}(\hat{g}_1), \phi_1^{-1}(\hat{h}_1)$, i.e, $\phi^{-1}(\hat{f}_1), \phi^{-1}(\hat{g}_1), \phi^{-1}(\hat{h}_1) \in K[x_1, \ldots, x_n]$. Since f and g just have trivial GCD, we must have that $\phi_1^{-1}(\hat{h}_1) \in \mathsf{K}$ and thus $\hat{h}_1 \in \mathsf{K}$.

Now consider the Sylvester resultant

$$\rho_1(\alpha_1, \ldots, \alpha_n) = \mathrm{Res}_X(\phi_1(f), \phi_1(g)) \in \mathsf{K}[\alpha_1, \ldots, \alpha_n].$$

Because $\mathrm{GCD}(\phi_1(f), \phi_1(g)) = 1$, even in $\mathsf{K}[X, \alpha_1, \ldots, \alpha_n]$, we have $\rho_1 \neq 0$. Now suppose that for $p_1, \ldots, p_n \in \mathsf{K}$ we have $\rho_1(p_1, \ldots, p_n) \neq 0$. First, we have $f_{1,1} \neq 0$ and $g_{1,1} \neq 0$, where $f_{1,1}$ and $g_{1,1}$ are defined in (3). We claim that $\mathrm{GCD}(f_{1,1}, g_{1,1}) = 1$. Here we need the condition on the B_i, since that condition guarantees that the leading coefficient $\lambda_1(B_2, \ldots, B_n)$ of $g_{1,1}$ is independent of $p_1, \ldots, p_n$ and therefore, considering the corresponding Sylvester matrices, we get

$$\mathrm{Res}_X(f_{1,1}, g_{1,1}) = \pm\frac{\rho_1(p_1, \ldots, p_n)}{\lambda_1(B_2, \ldots, B_n)^{\nu}} \neq 0,$$

where $\nu = \deg_X(\phi_1(f)) - \deg_X(f_{1,1})$, which establishes our claim.

The probability estimate for $t = 1$ now follows from the Schwartz-Zippel lemma [28, 24, 3] and the degree estimate $\deg(\rho_1) \le 2\deg(f)\deg(g)$.

Finally, we consider arbitrary J. As before, for $j \ge 1$ and $h \in \mathsf{K}[x_1, x_2, \ldots, x_n, \alpha_1]$ we introduce the map

$$\begin{aligned} \phi_j(h(x_1, x_2, \ldots, x_n, \alpha_1)) = h(X, B_2(X - \alpha_1^{\tau_{j,1}}) + \alpha_2^{\tau_{j,2}}, \ldots, \\ B_n(X - \alpha_1^{\tau_{j,1}}) + \alpha_n^{\tau_{j,n}}, \alpha_1^{\tau_{j,1}}). \end{aligned}$$

and the resultant

$$\rho_j(\alpha_1, \ldots, \alpha_n) = \mathrm{Res}_X(\phi_j(f), \phi_j(g)) \in \mathsf{K}[\alpha_1, \ldots, \alpha_n].$$

Now suppose that the leading coefficient in X of $\phi_1(f)$ is $\sigma \in \mathsf{K}[\alpha_1, \ldots, \alpha_n] \setminus \{0\}$. Then the leading coefficient in X of $\phi_j(f)$ is $\sigma(\alpha_1^{\tau_{j,1}}, \ldots, \alpha_n^{\tau_{j,n}})$, because the latter polynomial remains non-zero. Thus $\deg_X(\phi_j(f)) = \deg_X(\phi_1(f))$, $\rho_j(\alpha_1, \ldots, \alpha_n) = \rho_1(\alpha_1^{\tau_{j,1}}, \ldots, \alpha_n^{\tau_{j,n}}) \neq 0$ and

$$\begin{aligned} \mathrm{Res}_X(f_{1,j}, g_{1,j}) &= \pm\rho_j(p_1, \ldots, p_n)/\lambda_1(B_2, \ldots, B_n)^{\mu} \\ &= \pm\rho_1(p_1^{\tau_{j,1}}, \ldots, p_n^{\tau_{j,n}})/\lambda_1(B_2, \ldots, B_n)^{\mu}, \end{aligned}$$

where $\mu = \deg_X(\phi_j(f)) - \deg_X(f_{1,j})$. Therefore, any point $p_1, \ldots, p_n$ satisfies our lemma if $\prod_{j=1}^{J} \rho_1(p_1^{\tau_{j,1}}, \ldots, p_n^{\tau_{j,n}}) \neq 0$. The probability estimate follows from the degree estimate $\deg(\rho_1(\alpha_1^{\tau_{j,1}}, \ldots, \alpha_n^{\tau_{j,n}})) \le 2t\deg(f)\deg(g)$ for $j \ge 2$. □

We can now state our evaluation algorithm, which includes a method for determining the degrees of the numerator and denominator polynomials.

Algorithm *Evaluation of Numerator and Denominator*

Input:

- ► $\frac{f(x_1,x_2,\ldots,x_n)}{g(x_1,x_2,\ldots,x_n)} \in \mathsf{K}(x_1, x_2, \ldots, x_n)$ input as a black box (see above)
- ► $B_2, \ldots, B_n$: $n-1$ shift elements that are randomly chosen from a sufficiently large finite set $S_1 \subseteq \mathsf{K}$
- ► $p_1, \ldots, p_n$: n evaluation points that are randomly chosen from a sufficiently large finite set $S_2 \subseteq \mathsf{K}$
- ► $\bar{d}, \bar{e}$: degree bounds $\bar{d} \ge \deg(f)$ and $\bar{e} \ge \deg(g)$
- ► d, e (optional): the degrees of f and g, respectively (with high probability)
- ► $\tau_1, \ldots, \tau_n$: a given exponent vector with $1 \le \tau_i \le \min(\bar{d}, \bar{e})$

Output:

- ► the value of $f(p_1^{\tau_1}, \ldots, p_n^{\tau_n})/c$ and $g(p_1^{\tau_1}, \ldots, p_n^{\tau_n})$ $/c$ (with high probability), where c is the leading coefficient of $g(X, B_2X, \ldots, B_nX)$ (with high probability)
- ► or "failure," in which case the random values input are diagnosed as unusable

The algorithm performs a Cauchy interpolation (rational function recovery) for

$$f_{1,j}(X)/g_{1,j}(X) \bmod (X - i_1)\cdots(X - i_{d+e+1}),$$

where $f_{1,j}$ and $g_{1,j}$ are defined in (3) for $(\tau_{j,1}, \ldots, \tau_{j,n}) = (\tau_1, \ldots, \tau_n)$ and $i_l \in \mathsf{K}$ are suitable values. After making $g_{1,j}$ monic, the numerator and denominator values are $f_{1,j}(p_1^{\tau_1})$ and $g_{1,j}(p_1^{\tau_1})$. From Lemma 2.2, we know that $\mathrm{GCD}(f_{1,j}, g_{1,j}) = 1$ in $\mathsf{K}[X]$ and therefore the Cauchy interpolation algorithm recovers the proper images with high probability.

Case $\deg(f)$ and $\deg(g)$ are given:

EV1 Compute (possibly in parallel) $d+e+1$ distinct elements $i_1, \ldots, i_{d+e+1} \in \mathsf{K}$ and

$$A_l = \frac{f}{g}(i_l, B_2(i_l - p_1^{\tau_1}) + p_2^{\tau_2}, \ldots, B_n(i_l - p_1^{\tau_1}) + p_n^{\tau_n}) \neq \infty$$

$$\text{for all } 1 \le l \le d+e+1.$$

If $\deg(g_{1,j}) = \deg(g)$, i.e., the shift points $B_2, \ldots, B_n$ preserve the denominator degree, at most $d + 2e + 1$ elements in K need to be tried since there are at most e roots of $g_{1,j}(X)$.

If more than e values in K yield ∞ when evaluating the rational function black box return with "failure." Either the degrees are incorrect, or the projection points $B_2, \ldots, B_n$ and $p_1^{\tau_1}, \ldots, p_n^{\tau_n}$ are unlucky, or the black box does not evaluate a rational function.

EV2 By interpolation, compute a polynomial $h_1(X) \in \mathsf{K}[X]$ such that $h_1(i_l) = A_l$ for all $1 \le l \le d+e+1$ and $\deg(h_1) < d+e+1$.

EV3 By the extended Euclidean algorithm in Lemma 2.1 compute $\hat{g}, \hat{h}$ such that

$$\hat{h} \equiv \hat{g}h_1 \pmod{(X-i_1)\cdots(X-i_{d+e+1})}, \quad \deg(\hat{h}) \le d.$$

By construction we have $\deg(\hat{g}) \le e$. If $\deg(\hat{g}) < e$ then return "failure."

If $\mathrm{GCD}(\hat{g}, \hat{h}) \ne 1$ then return "failure." In this case, there is no rational function for the computed points (see [6, Corollary 5.18]), so again the degrees are incorrect or the black box does not evaluate a rational function.

EV4 Return $\hat{h}(p_1^{\tau_1})/c$ and $\hat{g}(p_1^{\tau_1})/c$ where c is the leading coefficient of $\hat{g}$.

Case $\bar{d} \ge \deg(f)$ and $\bar{e} \ge \deg(g)$ are given:
We determine the actual degrees by iterating on $k = d+e+1 = 1, 2, \ldots$ In the previous case, e is used to terminate the search for values i_l on which $g_{1,j}$ does not vanish. For this we use the bound $\bar{e}$ instead. The numerator degree d is used in Step EV3. Here we make the following change. First, we precompute for the threshold $\eta \ge 1$ the rational function values

$$U_m = \frac{f}{g}(u_m, B_2(u_m - p_1^{\tau_1}) + p_2^{\tau_2}, \ldots, B_n(u_m - p_1^{\tau_1}) + p_n^{\tau_n}) \ne \infty \quad \text{for all } 1 \le m \le \eta,$$

where u_m are uniformly randomly chosen from a sufficiently large finite subset $S_3 \subseteq \mathsf{K}$. Again, only $\eta + \bar{e}$ values are tried before reporting "failure." Then for each k we consider *all* remainder/co-factor pairs produced by the extended Euclidean algorithm, and which satisfy the degree bounds. A pair is accepted as $f_{1,j}/g_{1,j}$ if it satisfies the input degree bounds, co-primeness, and the corresponding fraction is equal to U_m when evaluating X at u_m, that for all $1 \le m \le \eta$. In addition to returning the numerator and denominator values as in Step EV4, we also return their degrees. The interpolant h_1 of Step EV2 can be incrementally computed from k to $k+1$ using Newton interpolation (the method of divided differences). Note that the iteration is terminated in failure if $k > \bar{d} + \bar{e}$, in which case the inputs are unlucky or wrong. □

We have the following probabilistic analysis for our algorithm. Suppose the above algorithm is called $J \ge 1$ times, using a single list of random shift elements $B_2, \ldots, B_n$, a single point $p_1, \ldots, p_n$ and the degrees d, e computed by the first call with $(\tau_1, \ldots, \tau_n) = (1, \ldots, 1)$ and correct degree bounds $\bar{d}, \bar{e}$. Then the algorithm does not return "failure" and the returned values are equal the values of f/c and g/c for all J calls with probability no less than

$\Big(1 - \dfrac{\deg(g)}{|S_1|}\Big)$ bounds the probability that $\lambda_1(B_2, \ldots, B_n) \ne 0$ (see Lemma 2.2)

$\times\Big(1 - \dfrac{2((J-1)t+1)\deg(f)\deg(g)}{|S_2|}\Big)$ bounds the probability that all points are usable (Lemma 2.2), conditional on the event that the shifts B_i work

$\times\Big(1 - \theta_2(d, e, \bar{d})\Big(\dfrac{\theta_1(d, e, \bar{d}, \bar{e})}{|S_3|}\Big)^{\eta}\Big)$, where θ_1 and θ_2 are defined below, bounds the probability that the correct degrees d, e are computed, conditional on good shifts and points. A wrong degree is returned if a false univariate continued fraction $\hat{h}/\hat{g}$ is accepted as $f_{1,j}/g_{1,j}$, that is we have

$$(\hat{h}g_{1,j} - f_{1,j}\hat{g})(u_m) = 0 \text{ for all } 1 \le m \le \eta. \tag{4}$$

The largest degrees which need to be considered are $\deg(\hat{h}) \le \min(\bar{d}, d+e)$ and $\deg(\hat{g}) \le \min(\bar{e}, d+e-1)$, the latter for the last false $k = d+e$. Now the left polynomial in (4) has degree no more than

$$\theta_1(d, e, \bar{d}, \bar{e}) = \max(\min(\bar{d}, d+e)+e, d+\min(\bar{e}, d+e-1))$$

so all u_m accept one false $\hat{h}/\hat{g}$ with probability no more than $(\theta_1(d, e, \bar{d}, \bar{e})/|S_3|)^{\eta}$. There are no more than $\theta_2(d, e, \bar{d}) = \sum_{k=1}^{d+e+1} \min(k, \bar{d}+1)$ such fraction candidates to be considered (for certain cases, one can lessen the bound using $\bar{e}$). The probability that at least one such event, namely acceptance of a false candidate, occurs is then bounded from above by the sum of the probabilities for each event.

3. EARLY TERMINATION IN SPARSE RATIONAL FUNCTION INTERPOLATION

We now describe the combination of the early termination version [13] of Zippel's [29] sparse multivariate interpolation algorithm with Algorithm Evaluation of Numerator and Denominator on page 205. Early termination is used to minimize the number of polynomial evaluations while keeping the size of the intermediate evaluation points small. Zippel's algorithm reconstructs a sparse polynomial, $h \in \mathsf{K}[x_1, \ldots, x_n]$ say, one variable at a time. A so-called anchor point $p_2, \ldots, p_n \in \mathsf{K}$ is chosen. For $i = 1, 2, \ldots, n$ the univariate images $\psi_{e_1,\ldots,e_{i-1}}(x_i, p_{i+1}, \ldots, p_n) \in \mathsf{K}[x_i]$ of the coefficients $\psi_{e_1,\ldots,e_{i-1}}(x_i, \ldots, x_n) \in \mathsf{K}[x_i, \ldots, x_n]$ of the non-zero terms $x_1^{e_1} \cdots x_{i-1}^{e_{i-1}}$ in h, viewed as a polynomial in $x_1, \ldots, x_{i-1}$ with coefficient in $\mathsf{K}[x_i, \ldots, x_n]$, are computed by interpolation from values $\psi_{e_1,\ldots,e_{i-1}}(b_i^{[\kappa]}, p_{i+1}, \ldots, p_n) \in \mathsf{K}$, where $b_i^{[\kappa]} \in \mathsf{K}$ for $\kappa = 1, 2, \ldots$ Those values are found from $h(p_1^{\tau}, \ldots, p_{i-1}^{\tau}, b_i^{[\kappa]}, p_{i+1}, \ldots, p_n)$ for $\tau = 0, 1, \ldots$ by solving a transposed Vandermonde system [2]. Zippel's [28] ingenious observation is that for random p_i any zero coefficient of $\psi_{e_1,\ldots,e_{i-1}}(x_i, p_{i+1}, \ldots, p_n)$ is with high probability the value of a zero polynomial, thus reducing the size of the transposed Vandermonde system to the number of non-zero terms at stage $i-1$. Díaz and Kaltofen [4] introduce a homogenizing variable x_0 and interpolate $\tilde{h}(x_0, x_1, \ldots, x_n) = h(x_0x_1, \ldots, x_0x_n)$. Then it is known from their degrees in x_0 and $x_1, \ldots, x_i$, respectively, that terms that do not depend on $x_{i+1}, \ldots, x_n$ are complete and need not be interpolated any further (are "permanently pruned"). Kaltofen and Lee [13] perform the interpolation of each $\psi_{e_1,\ldots,e_{i-1}}(x_i, p_{i+1}, \ldots, p_n)$ by "racing" both the early termination version of Newton interpolation and the early termination version of sparse univariate Ben-Or/Tiwari interpolation [14], that on the same evaluation points $b_i^{[\kappa]} = p_i^{\kappa}$. Then low degree or sparse $\psi_{e_1,\ldots,e_{i-1}}(x_i, p_{i+1}, \ldots, p_n)$ can be "temporarily" or permanently pruned from the interpolation problems at state i.

When combining the algorithm in [13] with Algorithm Evaluation of Numerator and Denominator on page 205 we can take further advantage of temporary pruning and early termination, namely when all terms of one of the numerator or denominator polynomials are completed (either temporarily or permanently). Because in that case, no univariate rational fraction recovery is needed for computing the values of the other remaining polynomial, and a single evaluation of the black box of the rational function suffices. We present a brief sketch of our algorithm.

Algorithm *Sparse Rational Function Interpolation*

Input:
- ▸ $\frac{f(x_1,x_2,\ldots,x_n)}{g(x_1,x_2,\ldots,x_n)} \in \mathsf{K}(x_1,x_2,\ldots,x_n)$ input as a black box
- ▸ $(x_1,\ldots,x_n)$: an ordered list of variables in f/g.
- ▸ $\bar{d},\bar{e}$: degree bounds $\bar{d} \geq \deg(f)$ and $\bar{e} \geq \deg(g)$

Output:
- ▸ $f(x_1,\ldots,x_n)/c$ and $g(x_1,\ldots,x_n)/c$ (with high probability), where $c \in \mathsf{K}$.
- ▸ Or "failure", in which case unlucky random elements have been selected (one can rerun the algorithm with new random values) or the black box does not evaluate a rational function of the given degree bounds.

ET1 Sample shift elements $B_2,\ldots,B_n$ randomly from a sufficiently large finite set $S_1 \subseteq \mathsf{K}$;

Initialize the anchor points: choose $p_0,p_1,\ldots,p_n$ randomly from a sufficiently large finite set $S_2 \subseteq \mathsf{K}$;

Introduce the homogenizing variable x_0 into f and g, define

$$\frac{\tilde{f}(x_0,x_1,\ldots,x_n)}{\tilde{g}(x_0,x_1,\ldots,x_n)} = \frac{f(x_0x_1,x_0x_2,\ldots,x_0x_n)}{g(x_0x_1,x_0x_2,\ldots,x_0x_n)}.$$

ET2 Interpolate Homogenizing Variable x_0:

Inputting the shift elements $B_2,\ldots,B_n$ and $\bar{d},\bar{e}$ to Algorithm Evaluation of Numerator and Denominator on page 205, compute evaluations of $\tilde{f}$ and $\tilde{g}$. The first such call returns degrees d,e that with high probability are the degrees of f and g. Note that for each evaluation one only needs $\deg(f)+\deg(g)+1$ black box probes.

With the obtained values, interpolate the polynomials $f_0 = \tilde{f}(x_0,p_1,\ldots,p_n)/c$ and $g_0 = \tilde{g}(x_0,p_1,\ldots,p_n)/c$, simultaneously using the racing algorithm described as above. Here c is the leading coefficient of the polynomial $g(X,B_2X,\ldots,B_nX)$.

If Algorithm Evaluation of Numerator and Denominator on page 205 or racing algorithm fail, then return "failure".

ET3 Interpolate Next Variable x_i:

Case $\tilde{f}(x_0,x_1,\ldots,x_n)/c$ or $\tilde{g}(x_0,x_1,\ldots,x_n)/c$ is completed:

The values of the yet-to-be complete polynomial is computed directly by the black box and the completed polynomial in place of Algorithm Evaluation of Numerator and Denominator on page 205, and a stage i sparse polynomial interpolation is performed as described above.

Case $\tilde{f}(x_0,x_1,\ldots,x_n)/c$ and $\tilde{g}(x_0,x_1,\ldots,x_n)/c$ are not completed:

Interpolate $f_i = \tilde{f}(x_0,x_1,\ldots,x_i,p_{i+1},\ldots,p_n)/c$ and $g_i = \tilde{g}(x_0,x_1,\ldots,x_i,p_{i+1},\ldots,p_n)/c$ simultaneously, which is similar to Step ET2. As in the previous case, the numerator or denominator may be completed early.

ET4 Recover $f(x_1,\ldots,x_n)/c$ and $g(x_1,\ldots,x_n)/c$ from f_n and g_n, respectively. This step is non-trivial for certain fields such as $\mathsf{K} = \mathbb{Q}$, when the scalar coefficients of both numerator and denominator can be reduced. See also Section 4. □

Note that our algorithm essentially performs simultaneous interpolation of two sparse polynomials, which are given by a black box that evaluates both at a given point. In our case, the black box operates differently when early termination has occurred, either temporarily or for the rest of the interpolation task. One can naturally generalize our techniques to interpolating an entire vector of multivariate sparse polynomials and rational functions. In the latter case, additional savings are possible (see the end of Section 4).

4. RATIONAL VECTOR RECOVERY

We now turn to the problem of recovering rational numbers from their modular images. The constructive version [15, Theorem 5.1] of Axel Thue's theorem establishes what is the corresponding integral property of the polynomials in Lemma 2.1.

THEOREM 4.1. *Let a residue $H \geq 1$, a modulus M, and bounds $D,E \geq 2$ be integers such that $H < M$, $(D-1)(E-1) < M < DE$. Then the problem*

$$F \equiv GH \pmod{M},\ |F| < D,\ F \neq 0,\ 0 < G < E \qquad (5)$$

is solvable in integers F,G if and only if $\Delta = \mathrm{GCD}(H,M) < D$. Furthermore, assuming that this is the case, let

$$\frac{U_0}{V_0} = \frac{0}{1}, \frac{U_1}{V_1}, \ldots, \frac{U_N}{V_N} = \frac{H/\Delta}{M/\Delta}, \quad V_N \geq \frac{M}{D-1} > E-1,$$

be the continued fraction approximations of H/M and choose l such that $V_l < E \leq V_{l+1}$. Then $G_1 = V_l$, $F_1 = HV_l - MU_l$ is a solution for (5). The set of all solutions for (5) exclusively either consists of λG_1, λF_1, where $1 \leq \lambda < \min(E/G_1, D/|F_1|)$ or else consists of G_1,F_1 and G_2,F_2 with $F_1F_2 < 0$. In the latter case we can determine G_2,F_2 from U_{l-1}/V_{l-1} or U_{l+1}/V_{l+1} in $O((\log M)^2)$ binary steps.

Note that D,E are bounds. In [15] examples for all three cases are given. If $\mathrm{GCD}(G,M) = 1$ then $F/G \equiv H \pmod{M}$ and a rational number F/G is recovered from its modulus. In modular arithmetic it is often known that such a solution exists. The exceptional case of two rational number candidates can be then resolved as in [27], by using a modulus M so that E is at least twice the denominator and selecting the solution with the smaller denominator as the recovered rational number, which is then F_1/G_1. If we choose the modulus even larger, the last denominator $V_l < E$ must then be substantially smaller than E and a large quotient must occur. In [21] this observation is used to determine l without E, assuming that the previous quotients in the continued fraction approximation are small.

We discuss simultaneous recovery $F_i/G \equiv H_i \pmod{M}$ for given $H_1,\ldots,H_t \in \mathbb{Z}_M$. Again we wish to determine l without E, while keeping M as small as possible.

Algorithm *Rational Vector Recovery 1*

Input:
- $M \geq 2$: a modulus; $H_1, \ldots, H_t \in \mathbb{Z}_M$
- s (optional): the range of small random residues; the number of random trials (optional)

Output:
- $G, D \in \mathbb{Z}_{\geq 2}$ that satisfy

$$\left.\begin{array}{l}\mathrm{GCD}(G,M)=1,\\ (D-1)G<M<D(G+1),\\ |GH_i \text{ smod } M|<D \text{ for all } 1\le i\le t,\end{array}\right\}\qquad (6)$$

where smod denotes the absolutely smallest remainder (symmetric residue).
- or "failure," in which case either the randomization was unlucky or no G, D that satisfy (6) exist.

For a given number of trials, repeat the following recovery procedure. Then return "failure."

VR1 Compute a random linear combination $H \equiv \gamma_1 H_1 + \cdots + \gamma_t H_t \in \mathbb{Z}_M$ where $-s \leq \gamma_i \leq s$ are uniformly randomly chosen. If $H = 0$ go to next trial.

VR2 For each continued fraction U_l/V_l where $l = 1, 2, \ldots$ of H/M perform the tests in Steps VR3 and VR4

VR3 Set $E \leftarrow V_l + 1$. If $\mathrm{GCD}(H, M) \geq E$ go to next trial. Compute the maximum bound D that satisfies $(D-1)(E-1) < M < DE$. Set G to G_1 and possibly G_2 as computed by Theorem 4.1. Note that for the second case in the proof of [15, Theorem 5.1], we currently assume the bound E. If $\mathrm{GCD}(G, M) > 1$ go to next value or trial.

VR4 Compute the maximum bound D that satisfies $(D-1)(E-1) < M < DE$. If $|GH_i \text{ smod } M| < D$ for all $1 \leq i \leq t$ return G, D. Otherwise go to next trial. □

Step VR1 is necessary because G is the least common multiple of the individual rational denominators. Our formulation of the problem is different from [22]. And our algorithm produces the first of potentially multiple solutions to (6). Any solution, including $H_1, \ldots, H_t$ and H_1 smod $M, \ldots, H_t$ smod M, i.e., $G = 1$ is a rational vector satisfying the congruences for certain bounds. Note that the case $H_1 = \cdots = H_t$ naturally leads to multiple rational solutions. For a given problem, a unique correct vector needs to be selected by other means. For the linear system problem [22], this can done by adjusting the bound D downward.

In test trials, the Algorithm Rational Vector Recovery 1 above performs unexpectedly well. Provided M is sufficiently large to accommodate the numerator and denominator sizes of the rational preimage and the size of the linear coefficients γ_i of Step VR1, the preimage is almost always returned. This is because false denominators are removed in the self-correction test VR4. However, if the least common denominator is substantially larger than the denominators of the individual components, a number of trials is sometimes needed.

It is possible to replace the scalar continued fraction approximation algorithm by a variant of the simultaneous diophantine approximation algorithm [20]. For a given vector $\alpha = (A_1/B_1, \ldots, A_t/B_t)$ and a given bound E one seeks a denominator G with $1 \leq G \leq E$ such that

$$\max_{1\le i\le t}\left\{\rho_i \mid \rho_i = \min_{\zeta_i\in\mathbb{Z}}\left|G\frac{A_i}{B_i}-\zeta_i\right|\right\}\qquad (7)$$

is minimized. Applying the minimization problem to $\alpha = (H_1/M, \ldots, H_t/M)$, one minimizes simultaneously $|GH_i - \zeta_i M|$, i.e., the numerators $F_i = \pm M\rho_i$ with $F_i \equiv GH_i \pmod{M}$. In [20] an algorithm, which iteratively performs several lattice basis reductions, is described that for the minimum distance $\rho_E^{[\min]}$ of (7) among any G in the range $1 \leq G \leq E$ computes a G^* with

$$1 \le G^* \le \sqrt{2^t}\cdot E \quad \text{and}$$

$$\max_{1\le i\le t}\left\{\rho_i^* \mid \rho_i^* = \min_{\zeta_i\in\mathbb{Z}}\left|G^*\frac{H_i}{M}-\zeta_i\right|\right\} \le \sqrt{5t\,2^{t-1}}\cdot\rho_E^{[\min]}.$$

One recovers $F_i^* = \pm M\rho_i^* \equiv G^* H_i \pmod{M}$. In order to keep t small, one can use several random linear combinations of the H_i instead of the entire vector.

Example 4.2. Consider the rational vector $\mathbf{V}$ and two different modular images $\mathbf{H} = \mathbf{V} \bmod M_1$ and $\bar{\mathbf{H}} = \mathbf{V} \bmod M_2$ with moduli $M_1 = 2^{25}$ and $M_2 = 2^{17}$ given in Figure 2.

$$\mathbf{V}=\begin{bmatrix}\frac{103}{5003}\\ \frac{1847}{5003}\\ -\frac{339}{5003}\\ -\frac{3772}{5003}\\ \frac{1060}{5003}\\ \frac{2234}{5003}\\ \frac{3085}{5003}\\ \frac{4826}{5003}\end{bmatrix},\ \mathbf{H}=\begin{bmatrix}19919381\\ 18718853\\ 12950951\\ 10677324\\ 25821420\\ 30361966\\ 127431\\ 16264142\end{bmatrix},\ \bar{\mathbf{H}}=\begin{bmatrix}127509\\ 106629\\ 105895\\ 60492\\ 236\\ 84334\\ 127431\\ 11214\end{bmatrix}.$$

Figure 2: Example vectors for rational recovery

Now we recover the vector $\mathbf{V}$ from the two images using both algorithms:

Case 1 $M_1 = 2^{25} = 33554432$. Applying Algorithm Rational Vector Recovery 1 on page 208 to $\mathbf{H}$, we need for $s = 5$ from 1 to 6 trials to recover $\mathbf{V}$. Using the simultaneous diophantine approximation algorithm with $E = \lceil\sqrt{M_1}\,\rceil$, we need a single lattice basis reduction to recover the rational numbers vector $\mathbf{V}$.

Case 2 $M_2 = 2^{17} = 131072$. We fail to recover $\mathbf{V}$ using Algorithm Rational Vector Recovery 1. However, we succeed to recover $\mathbf{V}$ with $E = \lceil\sqrt{M_2}\,\rceil$ after 5 iterations using our variant of the simultaneous diophantine approximation algorithm. □

The problem of rational vector recovery of course applies also to our sparse rational function interpolation problem. Like in Algorithm Evaluation of Numerator and Denominator on page 205 and Algorithm Sparse Rational Function Interpolation on page 207, for interpolating *a vector* of sparse rational functions with common denominator we can employ simultaneous recovery of univariate fractions F_j/G from their modular images H_j with

$$F_j \equiv G\,H_j \pmod{(X-i_1)\cdots(X-i_\kappa)}.$$

Olesh and Storjohann [22] show that for a number of points κ less than $d+e+1$ fractions of numerator degree d and denominator degree e can be recovered in certain cases, now by a minimal polynomial basis algorithm. Thus the combined number of black box evaluations is reduced.

5. EXPERIMENTS

Algorithm Sparse Rational Function Interpolation on page 207 has been implemented in Maple. We report the results of the experiments using our algorithm which are shown in Table 1 below. For each example, we construct two relatively prime polynomials with random integer coefficients in the given range as the numerator and denominator. Here *Coeff. Range* is the range of the coefficients of the numerator and the denominator; d_f and d_g are the degrees of the numerator and the denominator of the rational function respectively; n denotes the number of the variables of the rational function; t_f and t_g are the number of the terms of the numerator and the denominator respectively; mod is the integer of the modulus; N_1 denotes the number of the evaluations to interpolate the rational function; N_2 denotes the number of the black box probes to interpolate the rational function. In all cases, for Algorithm Evaluation of Numerator and Denominator we use a threshold value $\eta = 3$ in the first call.

Ex.	*Coeff. Range*	d_f, d_g	n	t_f, t_g	mod	N_1	N_2
1	[-10,10]	3, 3	2	6, 6	503	31	221
2	[-10,10]	5, 2	4	6, 3	1009	65	339
3	[-20,20]	2, 4	6	2, 5	120011	62	357
4	[-20,20]	1, 6	8	4, 8	8009	141	777
5	[-30,30]	10, 5	10	7, 4	4001	164	2246
6	[-10,10]	15, 15	15	15, 15	50021	555	17120
7	[-10,10]	20, 20	20	20, 20	50021	968	38682
8	[-30,30]	30, 15	5	20, 10	10007	326	12896
9	[-10,10]	100, 60	20	100, 60	1000003	5597	873843
10	[-50,50]	50, 50	50	50, 50	1000003	6075	603638
11	[-10,10]	2, 8	90	10, 50	1000003	7135	75082

Table 1: Algorithm performance on benchmarks

6. SPARSE NUMERICAL INTERPOLATION OF RATIONAL FUNCTIONS (SNIPR)

In [7] a numerical algorithm is given to interpolate the sparse multivariate polynomial from a multivariate approximate black-box polynomial, making use of approximate evaluations at random primitive roots of unity. In order to interpolate the approximate sparse rational functions from the black box with noisy outputs, it is necessary to compute the numerator and denominator evaluations at a random primitive roots of unity.

In the exact case, the univariate rational function can be recovered by padé approximation. From Lemma 2.1 we know that the degree of the numerator and denominator can be determined by extended Euclidean schemes when the bound of the rational function is given. However, in the approximate case, the polynomial H in Lemma 2.1 is not exact because of the approximate black box. So it is difficult to determine the degrees of F and G, i.e. the degrees of the numerator and denominator. It is a remaining problem we have not completely addressed. For simplicity, we assume that the degree of rational function is known. In section 2, our exact algorithm performs a Cauchy interpolation for

$$f_{1,j}(X)/g_{1,j}(X) \bmod (x - i_1) \cdots (x - i_{d+e+1}),$$

where $f_{1,j}, g_{1,j}$ are defined in (3). After making $g_{1,j}$ monic, the numerator and denominator values are $f_{1,j}(p_1^j)$, $g_{1,j}(p_1^j)$.

Now we describe our method to compute the numerical evaluation of the numerator and denominator in detail. According to our exact algorithm and the numerical algorithm for multivariate polynomial interpolation in [7], we choose the shift elements and variable values to be the roots of unity, namely $B_j = \exp(2s_j\pi\boldsymbol{i}/b_j) \in \mathbb{C}$ ($\boldsymbol{i} = \sqrt{-1}$ and $2 \le j \le n$) and $p_k = \exp(2s_{n+k}\pi\boldsymbol{i}/b_{n+k}) \in \mathbb{C}$ ($1 \le k \le n$) where $b_2, \ldots, b_{2n} \in \mathbb{Z}_{>0}$ are pairwise relatively prime such that $b_l > \max(d, e)$ (d, e the numerator and denominator degrees) and where s_l are random integers with $1 \le s_l < b_l$. In order to recover the univariate polynomials $f_{1,j}, g_{1,j}$, in place of extended Euclidean schemes we apply to solve the Toeplitz-like linear system like Example on page 302 in [11]:

$$w(x)h_0(X) + g_{1,j}(X)h_1(X) - f_{1,j}(X) = 0 \qquad (8)$$

where $h_0(X) = (X - i_1)\cdots(X - i_{d+e+1})$, $h_1(X)$ is a interpolant such that $h_1(i_k) = f_{1,j}(i_k)/g_{1,j}(i_k)$ for all $1 \le k \le d + e + 1$, and the degrees of $f_{1,j}, g_{1,j}$ are d, e respectively. From the equation (8) we get a $(2d + e + 1) \times (2d + e + 2)$ matrix called M. Since the row dimension of M is one less than the column dimension M. The system (8) always have a solution. $f_{1,j}, g_{1,j}$ are obtained from the null space of M. Then we get the numerator and denominator values from $f_{1,j}, g_{1,j}$.

In order to obtain a better solution, we can oversample at $d + e + 1 + \zeta$ points, where $\zeta \ge 1$, compute the polynomials $h_0(X)$ and $h_1(X)$, and then compute the approximate solution $\mathbf{x}$ of the overdetermined system: $M \cdot \mathbf{x} \approx \mathbf{0}$. This problem is a structured total squares problem since the matrix A has a Toeplitz-block structure (cf. [9]). We apply the *Structured Total Least Norm (STLN)* [23] method to obtain the approximate solution. As [19, 17] described, $\mathbf{b}$ can be chosen a column of M and A are formed by the remaining columns. We seek to compute the minimal structure preserving perturbation $\mathbf{h}, E$ such that $(A + E) \cdot \mathbf{x} = \mathbf{b} + \mathbf{h}$. Then we obtain the coefficients of univariate numerator and denominator from $\mathbf{x}$.

Example 6.1. Consider the rational function f/g:

$$f = x^3 + y^3 + 3x\,y + 4x + 1 \quad \text{and} \quad g = 3x^3 + 2xy^2 + 5xy + 4x + 5.$$

The noise for the black box of f/g is in the range of $10^{-9} \approx 10^{-7}$. Choose $p_1 = \exp(2\pi\boldsymbol{i}/5), p_2 = \exp(2\pi\boldsymbol{i}/11)$ and $B_2 = \exp(2\pi\boldsymbol{i}/13)$. We seek to compute the approximate evaluation of the numerator and denominator: $f_{1,j}(p_1^j), g_{1,j}(p_1^j)$, $1 \le j \le 9$. We use STLN method to solve the overdetermined system and then obtain two lists of the values of the numerator and denominator $C_j, D_j, 1 \le j \le 9$. Now we check the backward error of our evaluation:

$$\sum_{j=1}^{9} \|C_j - f(p_1^j)/c\|^2 + \|D_j - g(p_1^j)/c\|^2 = 3.45097 \times 10^{-11}$$

where $c = 4.13613 + 1.64597i$ is the leading coefficient of the polynomial $g(X, B_2X)$.

Using the algorithm in [7] the approximate numerator and denominator is interpolated according to the above evaluation C_j, D_j:

$$\begin{aligned}\tilde{f} &= (0.20872 - 0.08306\boldsymbol{i})y^3 + (0.20872 - 0.08305\boldsymbol{i})x^3\\ &\quad + (0.62616 - 0.24918\boldsymbol{i})x\,y + 0.83487 - 0.33224\boldsymbol{i})x\\ &\quad + 0.20872 - 0.08306\boldsymbol{i},\\ \tilde{g} &= (0.62616 - 0.24918\boldsymbol{i})x^3 + (0.41744 - 0.16612\boldsymbol{i})xy^2\\ &\quad + (1.04359 - 0.41529\boldsymbol{i})xy + (0.83487 - 0.33224\boldsymbol{i})x\\ &\quad + 1.04356 - 0.41529\boldsymbol{i}.\end{aligned}$$

The backward error is $\|\tilde{f} - f/c\|_2^2 + \|\tilde{g} - g/c\|_2^2 = 5.08936 \times 10^{-14}$. □

In the exact case, we require that the polynomials $f_{1,j}$ and $g_{1,j}$ are relatively prime. Now one approach is to check whether $f_{1,j}$ and $g_{1,j}$ have an approximate GCD. First, for the given map and the input degrees of the rational function (d, e) we use our STLN method to compute $f_{1,j}, g_{1,j}$ from (8), and compute the backward error:

$$error_1 = \|w(X)h_0(X) + g_{1,j}(X)h_1(X) - f_{1,j}(X)\|.$$

Then decreasing the input degrees as $(d-1, e-1)$, we construct the overdetermined system from (8), where the degrees of $\hat{f}_{1,j}$, $\hat{g}_{1,j}$ are $d-1$, $e-1$ respectively. Then we compute $\hat{f}_{1,j}$ and $\hat{g}_{1,j}$ and compute the backward error:

$$error_2 = \|\hat{w}(X)h_0(X) + \hat{g}_{1,j}(X)h_1(X) - \hat{f}_{1,j}(X)\|.$$

Suppose the ratio $\Upsilon = error_2/error_1$ is sufficient large, that is $\Upsilon > \varepsilon$ where ε is a chosen large value. We can declare that $f_{1,j}$ and $g_{1,j}$ have no approximate GCD, that is $f_{1,j}(p_1^j)$ and $f_{1,j}(p_1^j)$ are the approximate evaluation $f(p_1^j, \dots p_n^j)/c$ and $f(p_1^j, \dots p_n^j)/c$. Otherwise, $f_{1,j}$ and $g_{1,j}$ have a nearby approximate GCD. Then we start fresh and select new $B_2, \dots, B_n$ or new $p_1, \dots, p_n$ to construct the new map. Therefore, we need to find $B_2, \dots, B_n$ and $p_1, \dots, p_n$, and compute $f_{1,j}, g_{1,j}$ such that they are relatively prime for all $1 \le j \le J$. In the exact case, from Lemma 2.2 we know that $f_{1,j}$ and $g_{1,j}$ are relatively prime with high probability for all $1 \le j \le J$. In the approximate case it seems difficult to have J consecutive approximately relatively prime pairs $f_{1,j}$ and $g_{1,j}$. We have overcome those difficulties by performing Zippel's sparse interpolation method [29] directly on sparse rational functions with noisy values [18].

Acknowledgement: We thank Wen-shin Lee for providing her numeric sparse interpolation code to us, Arne Storjohann for sending us [22] and discussions on rational vector recovery, and Lihong Zhi for discussions on numerical univariate rational function recovery.

7. REFERENCES

[1] Becuwe, S., Cuyt, A., and Verdonk, B. Multivariate rational interpolation of scattered data. In *Large-Scale Scientific Computing* (Heidelberg, Germany, 2004), I. Lirkov, S. Margenov, J. Wasniewski, and Y. Plamen, Eds., vol. 2907 of *Lect. Notes Comput. Sci.*, Springer Verlag, pp. 204–213.

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

[3] DeMillo, R. A., and Lipton, R. J. A probabilistic remark on algebraic program testing. *Information Process. Letters 7*, 4 (1978), 193–195.

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

[5] Dumas, J.-G., Ed. *ISSAC MMVI Proc. 2006 Internat. Symp. Symbolic Algebraic Comput.* (New York, N. Y., 2006), ACM Press.

[6] von zur Gathen, J., and Gerhard, J. *Modern Computer Algebra.* Cambridge University Press, Cambridge, New York, Melbourne, 1999. Second edition 2003.

[7] Giesbrecht, M., Labahn, G., and Lee, W. Symbolic-numeric sparse interpolation of multivariate polynomials. In Dumas [5], pp. 116–123.

[8] Grigoriev, D. Y., Karpinski, M., and Singer, M. F. Computational complexity of sparse rational function interpolation. *SIAM J. Comput. 23* (1994), 1–11.

[9] Kai, H. Rational function approximation and its ill-conditioned property. In Wang and Zhi [26], pp. 47–53. Preliminary version in [25], pp. 62–64.

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

[11] Kaltofen, E. Asymptotically fast solution of Toeplitz-like singular linear systems. In *Proc. 1994 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'94)* (New York, N. Y., 1994), ACM Press, pp. 297–304. Journal version in [12].

[12] Kaltofen, E. Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems. *Math. Comput. 64*, 210 (1995), 777–806.

[13] Kaltofen, E., and Lee, W. Early termination in sparse interpolation algorithms. *J. Symbolic Comput. 36*, 3–4 (2003), 365–400. Special issue Internat. Symp. Symbolic Algebraic Comput. (ISSAC 2002). Guest editors: M. Giusti & L. M. Pardo.

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

[15] Kaltofen, E., and Rolletschek, H. Computing greatest common divisors and factorizations in quadratic number fields. *Math. Comput. 53*, 188 (1989), 697–720.

[16] 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.

[17] Kaltofen, E., Yang, Z., and Zhi, L. Approximate greatest common divisors of several polynomials with linearly constrained coefficients and singular polynomials. In Dumas [5], pp. 169–176. Full version, 21 pages. Submitted, December 2006.

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

[19] Kaltofen, E., Yang, Z., and Zhi, L. Structured low rank approximation of a Sylvester matrix. In Wang and Zhi [26], pp. 69–83. Preliminary version in [25], pp. 188–201.

[20] Lagarias, J. C. The computational complexity of simultaneous diophantine approximation problems. *SIAM J. Comp. 14* (1985), 196–209.

[21] Monagan, M. Maximal quotient rational reconstruction: An almost optimal algorithm for rational reconstruction. In *ISSAC 2004 Proc. 2004 Internat. Symp. Symbolic Algebraic Comput.* (New York, N. Y., 2004), J. Gutierrez, Ed., ACM Press, pp. 243–249.

[22] Olesh, Z., and Storjohann, A. The vector rational function reconstruction problem, Sept. 2006. Manuscript, 14 pages.

[23] Park, H., Zhang, L., and Rosen, J. B. Low rank approximation of a Hankel matrix by structured total least norm. *BIT 39*, 4 (1999), 757–779.

[24] Schwartz, J. T. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM 27* (1980), 701–717.

[25] Wang, D., and Zhi, L. Eds. *Internat. Workshop on Symbolic-Numeric Comput. SNC 2005 Proc.* (2005). Distributed at the Workshop in Xi'an, China, July 19–21.

[26] Wang, D., and Zhi, L., Eds. *Symbolic-Numeric Computation.* Trends in Mathematics. Birkhäuser Verlag, Basel, Switzerland, 2007.

[27] Wang, P. S., Guy, M. J. T., and Davenport, J. H. P-adic reconstruction of rational numbers. *SIGSAM Bulletin 16*, 2 (May 1982), 2–3.

[28] Zippel, R. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation* (Heidelberg, Germany, 1979), vol. 72 of *Lect. Notes Comput. Sci.*, Springer Verlag, pp. 216–226. Proc. EUROSAM '79.

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