Theoretical Computer Science 412 (2011) 1445–1456

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Sparse interpolation of multivariate rational functions

Annie Cuyt *, Wen-shin Lee

Department of Mathematics and Computer Science, University of Antwerp, Middelheimlaan 1, 2020 Antwerpen, Belgium

ARTICLE INFO

Keywords:
Rational function
Sparse interpolation
Multivariate interpolation
Sparse polynomial
Black box

ABSTRACT

Consider the black box interpolation of a τ-sparse, n-variate rational function f, where τ is the maximum number of terms in either numerator or denominator. When numerator and denominator are at most of degree d, then the number of possible terms in f is $O(d^n)$ and explodes exponentially as the number of variables increases. The complexity of our sparse rational interpolation algorithm does not depend exponentially on n anymore. It still depends on d because we densely interpolate univariate auxiliary rational functions of the same degree. We remove the exponent n and introduce the sparsity τ in the complexity by reconstructing the auxiliary function's coefficients via sparse multivariate interpolation.

The approach is new and builds on the normalization of the rational function's representation. Our method can be combined with probabilistic and deterministic components from sparse polynomial black box interpolation to suit either an exact or a finite precision computational environment. The latter is illustrated with several examples, running from exact finite field arithmetic to noisy floating point evaluations. In general, the performance of our sparse rational black box interpolation depends on the choice of the employed sparse polynomial black box interpolation. If the early termination Ben-Or/Tiwari algorithm is used, our method achieves rational interpolation in $O(\tau d)$ black box evaluations and thus is sensitive to the sparsity of the multivariate f.

1. Introduction

The sparse interpolation of a black box multivariate rational function

$$f(x_1,\ldots,x_n)=\frac{p(x_1,\ldots,x_n)}{q(x_1,\ldots,x_n)}$$

where $p(x_1,\ldots,x_n)$ and $q(x_1,\ldots,x_n)$ are multivariate polynomials of at most total degree d, is to reconstruct f in a cost that is sensitive to its sparsity, namely the number of non-zero terms in p and q in the power basis, instead of its dense representation size $O(d^n)$, which explodes exponentially in the number of variables n. The black box representation of f outputs the evaluation of f for any given input.

Although univariate rational interpolation has been extensively studied, its multivariate counterpart dates from the last decades and is relatively new. Moreover, most of the work has been done with respect to dense interpolation, where it is assumed that all terms up to a certain degree are present in the numerator and denominator polynomial. With respect to so-called multivariate sparse rational interpolation, few results have been obtained in a handful of papers [13,11,10,5,19,20]. Especially in the context of floating point arithmetic, where the presence of round-off errors complicates matters,

* Corresponding author.
E-mail addresses: annie.cuyt@ua.ac.be (A. Cuyt), wen-shin.lee@ua.ac.be (W.-s. Lee).
URLs: http://www.cant.ua.ac.be (A. Cuyt), http://www.wen-shin.com (W.-s. Lee).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2010.11.050

multivariate sparse rational interpolation can be improved. We summarize the state of the art implementations in both exact and floating point arithmetic.

In [15,18] a method is developed to separately evaluate the numerator and denominator of a black box rational function. Then in [19] a probabilistic approach for sparse multivariate rational interpolation is outlined from a combination of [18] with the sparse interpolation algorithms in [16,8] to simultaneously interpolate numerator and denominator from their evaluations. However, it fails numerically on large degree inputs [19, Section 6]. An alternative is to use the extension of Zippel's probabilistic sparse interpolation to multivariate rational functions [20], which proceeds with one variable at a time and interpolates densely.

We introduce a new approach which makes use of the normalization [1,3] of the form in which the multivariate rational function is represented: one of the coefficients in numerator or denominator can be fixed to make the representation in a certain basis unique. From the multivariate rational function a set of auxiliary univariate rational functions is obtained and interpolated densely. The sparsity of the multivariate function is preserved in the coefficients of the auxiliary functions. We combine this with a choice of techniques from sparse polynomial interpolation, making our algorithm applicable in either exact or floating point arithmetic depending on the chosen technique. The multivariate rational function is then reconstructed from the coefficients of the auxiliary functions through sparse interpolation. Hence we can exploit the sparsity in the coefficients as in [19] and maintain sufficient numerical robustness as in [20].

The number of black box probes required for interpolation in our method still depends on the total degree d because each auxiliary univariate function is interpolated densely. But by using multivariate sparse polynomial interpolation to recover the coefficients of the auxiliary function, we are able to bring down the $O(d^n)$ complexity of the dense case and exploit the sparsity of the rational function in the multivariate case.

The overall performance of our rational interpolation depends on the choice of sparse polynomial interpolation plugged in. For example, if we use the early termination Ben-Or/Tiwari algorithm [16] or its symbolic-numeric variant [9], our rational interpolation requires $O(\tau d)$ black box evaluations. On the other hand, if Zippel's variable by variable interpolation [26] or its numerical variant [20] is chosen, then $O(n\tau d^2)$ evaluations are needed for the multivariate rational interpolation.

In Section 2 we present the basic idea, first for a special case and then in general. In Section 3 we illustrate the implementation in either exact or floating point arithmetic. The illustrations range from exact finite field arithmetic to noisy floating point evaluations.

2. Multivariate sparse rational interpolation

If a given black box multivariate rational function is defined at $(0, \ldots, 0)$, its constant term in the denominator is known to be non-zero and can be normalized to 1. In such a case our approach to sparse multivariate rational interpolation is fairly straightforward. We explain it in Section 2.1.

In general, a black box multivariate rational function may not be defined at $(0, \ldots, 0)$. But one can always choose a shifted power basis such that the rational function has a non-zero constant in the denominator. However, the sparsity of the original multivariate rational function may be lost in this representation. In Section 2.2 we extend the sparse interpolation algorithm of Section 2.1 to a shifted power basis representation, such that it remains sensitive to the original multivariate sparsity.

All the discussions are stated for a general field $\mathbb{K}$, as in a polynomial ring $\mathbb{K}[x_1, \ldots, x_n]$ or a rational field $\mathbb{K}(x_1, \ldots, x_n)$. The implementations and corresponding issues arising in various arithmetic environments, including exact and floating point, are addressed in Section 3.

2.1. Normalized multivariate rational functions

Suppose a black box multivariate rational function f, expressed in the multinomial basis, is defined at $(0, \ldots, 0)$. Since the denominator's constant term is non-zero, f can be normalized such that the constant in its denominator is 1. In other words, f can be written as

$$f(x_1, \ldots, x_n) = \frac{\overbrace{a_1 x_1^{d_{1,1}} \cdots x_n^{d_{1,n}} + \cdots + a_s x_1^{d_{s,1}} \cdots x_n^{d_{s,n}}}^{p(x_1,\ldots,x_n)}}{\underbrace{1 + b_2 x_1^{e_{2,1}} \cdots x_n^{e_{2,n}} + \cdots + b_t x_1^{e_{t,1}} \cdots x_n^{e_{t,n}}}_{q(x_1,\ldots,x_n)}}, \tag{1}$$

where $a_k \neq 0$, $b_\ell \neq 0$ and $p(x_1, \ldots, x_n), q(x_1, \ldots, x_n) \in \mathbb{K}[x_1, \ldots, x_n]$. We further require p and q to be relatively prime.

By introducing the homogenizing variable z to f as in [7,17], we form an auxiliary rational function $F(z, x_1, \ldots, x_n)$,

$$F(z, x_1, \ldots, x_n) = f(x_1 z, \ldots, x_n z) = \frac{\overbrace{A_0(x_1, \ldots, x_n) \cdot z^0 + A_1(x_1, \ldots, x_n) \cdot z + \cdots + A_\nu(x_1, \ldots, x_n) \cdot z^\nu}^{P(z) \in \mathbb{K}[x_1,\ldots,x_n][z]}}{\underbrace{1 + B_1(x_1, \ldots, x_n) \cdot z + \cdots + B_\delta(x_1, \ldots, x_n) \cdot z^\delta}_{Q(z) \in \mathbb{K}[x_1,\ldots,x_n][z]}}. \tag{2}$$

In (2) terms are collected with respect to the homogenizing variable z. The numerator and denominator, denoted by $P(z)$ and $Q(z)$ respectively, are regarded as univariate polynomials in z with coefficients from $\mathbb{K}[x_1, \ldots, x_n]$. The degrees of $P(z)$ and $Q(z)$ are ν and δ. They are also the respective total degrees of $p(x_1, \ldots, x_n)$ and $q(x_1, \ldots, x_n)$ in f given in (1).

The coefficients in $P(z)$ and $Q(z)$ are multivariate polynomials

$$A_k(x_1, \ldots, x_n) = \sum_{d_{j,1}+\cdots+d_{j,n}=k} a_j x_1^{d_{j,1}} \cdots x_n^{d_{j,n}}, \quad 0 \le k \le \nu,$$

$$B_0 = 1, \qquad B_\ell(x_1, \ldots, x_n) = \sum_{e_{j,1}+\cdots+e_{j,n}=\ell} b_j x_1^{e_{j,1}} \cdots x_n^{e_{j,n}}, \quad 1 \le \ell \le \delta,$$

and together these coefficients A_k and B_ℓ collect all the non-zero terms in p and q,

$$p(x_1, \ldots, x_n) = \sum_{k=0}^{\nu} A_k(x_1, \ldots, x_n), \qquad q(x_1, \ldots, x_n) = \sum_{\ell=0}^{\delta} B_\ell(x_1, \ldots, x_n). \tag{3}$$

If we interpolate A_k for $0 \le k \le \nu$ and B_ℓ for $1 \le \ell \le \delta$, then both p and q and hence $f = p/q$ can be determined.

To interpolate A_k and B_ℓ, we need to evaluate them at some chosen points $(\omega_1^{(i)}, \ldots, \omega_n^{(i)})$. Assume that the total degrees ν and δ are given. For a fixed $(\omega_1, \ldots, \omega_n)$, we consider the interpolation of $F(z, \omega_1, \ldots, \omega_n) = f(\omega_1 z, \ldots, \omega_n z)$ as a function of z. The function $F(z, \omega_1, \ldots, \omega_n)$ can be revealed by a (dense) univariate rational interpolation from its evaluation at distinct values $\zeta_0, \zeta_1, \ldots, \zeta_{\nu+\delta}$ for z,

$$f(\omega_1\zeta_0, \ldots, \omega_n\zeta_0), f(\omega_1\zeta_1, \ldots, \omega_n\zeta_1), \ldots, f(\omega_1\zeta_{\nu+\delta}, \ldots, \omega_n\zeta_{\nu+\delta}).$$

Once $F(z, \omega_1, \ldots, \omega_n)$ is interpolated, we simultaneously have at our disposal the evaluations $(\omega_1, \ldots, \omega_n)$ of the coefficients A_k and B_ℓ in

$$F(z, \omega_1, \ldots, \omega_n) = \frac{A_0(\omega_1, \ldots, \omega_n) \cdot z^0 + \cdots + A_\nu(\omega_1, \ldots, \omega_n) \cdot z^\nu}{1 + B_1(\omega_1, \ldots, \omega_n) \cdot z + \cdots + B_\delta(\omega_1, \ldots, \omega_n) \cdot z^\delta}. \tag{4}$$

Hence A_k and B_ℓ can be interpolated in parallel. If we use an early termination sparse algorithm [17,16], each of the simultaneous interpolations of A_k and B_ℓ is correct with high probability and reflects the sparsity of the corresponding A_k or B_ℓ. Since each of the A_k and B_ℓ collects a part of the non-zero terms in p and q as in (3), its sparsity is bounded by the overall sparsity of either p or q. The choice of evaluation points $(\omega_1^{(0)}, \ldots, \omega_n^{(0)})$, $(\omega_1^{(1)}, \ldots, \omega_n^{(1)})$, $\ldots$ for $(\omega_1, \ldots, \omega_n)$ depends on the interpolation algorithm employed, e.g. Zippel's probabilistic interpolation [26] or the early termination Ben-Or/Tiwari algorithm [17]. In the case of floating point arithmetic, we recommend a qd-scheme for the sparse polynomial interpolation [4].

Algorithm: `Sparse Rational Interpolation <normalized>`

Input:
- $f(x_1, \ldots, x_n)$: a multivariate black box rational function.
- ν and δ: total degrees of the numerator p and denominator q.
- η: a positive integer (or default to 1), the threshold required by the early termination strategy [16].

Output:
- $a_k, b_\ell, (d_{k,1}, \ldots, d_{k,n}), (e_{\ell,1}, \ldots, e_{\ell,n})$:

$$f(x_1, \ldots, x_n) = \frac{\sum_{k=1}^{s} a_k x_1^{d_{k,1}} \cdots x_n^{d_{k,n}}}{1 + \sum_{\ell=2}^{t} b_j x_1^{e_{\ell,1}} \cdots x_n^{e_{\ell,n}}} \text{ with high probability,}$$

or an error message if the procedure fails to complete [16].

Steps:

1. [Homogenization.]
   For $i = 0, 1, 2, \ldots$:
   Generate $(\omega_1^{(i)}, \ldots, \omega_n^{(i)})$. The choice of $(\omega_1^{(i)}, \ldots, \omega_n^{(i)})$ depends on the sparse polynomial interpolation algorithm to be employed.
2. [Dense univariate rational interpolation.]
   Pick distinct $\zeta_0, \zeta_1, \ldots, \zeta_{\nu+\delta}$ and evaluate $f(\omega_1^{(i)}\zeta_j, \ldots, \omega_n^{(i)}\zeta_j)$ for $0 \le j \le \nu+\delta$. From the $\nu+\delta+1$ evaluations interpolate

$$\begin{aligned} F(z, \omega_1^{(i)}, \ldots, \omega_n^{(i)}) &= f(\omega_1^{(i)} z, \ldots, \omega_n^{(i)} z) \\ &= \frac{A_0(\omega_1^{(i)}, \ldots, \omega_n^{(i)}) \cdot z^0 + \cdots + A_\nu(\omega_1^{(i)}, \ldots, \omega_n^{(i)}) \cdot z^\nu}{1 + B_1(\omega_1^{(i)}, \ldots, \omega_n^{(i)}) \cdot z + \cdots + B_\delta(\omega_1^{(i)}, \ldots, \omega_n^{(i)}) \cdot z^\delta}. \end{aligned} \tag{5}$$

3. [Simultaneous sparse multivariate polynomial interpolations.]
   The coefficients in (5) are the evaluations

$$A_0(\omega_1^{(i)},\ldots,\omega_n^{(i)}), A_1(\omega_1^{(i)},\ldots,\omega_n^{(i)}),\ldots,A_\nu(\omega_1^{(i)},\ldots,\omega_n^{(i)}),$$
$$B_1(\omega_1^{(i)},\ldots,\omega_n^{(i)}),\ldots,B_\delta(\omega_1^{(i)},\ldots,\omega_n^{(i)}).$$

   Use the early termination sparse algorithm to continue the simultaneous interpolations of A_k and B_ℓ for $0 \le k \le \nu$ and $1 \le \ell \le \delta$.

   If all A_k and B_ℓ are interpolated via early termination, then break out of the i loop.

For each i, the univariate (dense) rational interpolation of $F(z,\omega_1^{(i)},\ldots,\omega_n^{(i)})$ in (5) needs $O(\nu+\delta)$ evaluations. Each coefficient polynomial A_k or B_ℓ, collects a subset of terms in p or q. Its sparsity is bounded by the overall sparsity of p or q, which also bounds the i loop via the early termination of the employed sparse interpolation algorithm.

2.2. General multivariate rational functions

In Section 2.1, we require the multivariate rational function to have a non-zero constant in the denominator. That condition cannot be assumed in general. The purpose of the non-zero constant is to guarantee an a priori normalization of f. We can choose another non-zero term in f for normalization, but we still need the knowledge of such a non-zero term beforehand.

By shifting the power basis, we can always impose a non-zero constant in the denominator. However, a sparse polynomial or rational function representation may become dense after the basis is shifted. Here we demonstrate a shifting strategy that can preserve the sparse interpolation of f.

From here on we do not impose any restriction on the constant term of q in $f = p/q$. The numerator and denominator of f are denoted by

$$\begin{aligned} p(x_1,\ldots,x_n) &= \sum_{k=1}^{s} a_k x_1^{d_{k,1}}\cdots x_n^{d_{k,n}}, \quad a_k \neq 0,\\ q(x_1,\ldots,x_n) &= \sum_{\ell=1}^{t} b_\ell x_1^{e_{\ell,1}}\cdots x_n^{e_{\ell,n}}, \quad b_\ell \neq 0. \end{aligned} \tag{6}$$

Let us introduce a shifted homogenization of f and form another auxiliary function $\Gamma(z,x_1,\ldots,x_n)$. Recall that in Section 2.1, the normalization of f directly leads to the normalization of the auxiliary function F. Now we normalize the shifted auxiliary function Γ instead of f.

For any $(\sigma_1,\ldots,\sigma_n)$ where $f = p/q$ is defined, $q(\sigma_1,\ldots,\sigma_n) \neq 0$. So we define the $(\sigma_1,\ldots,\sigma_n)$-shifted homogenization of f and the auxiliary function

$$\begin{aligned} \Gamma(z,x_1,\ldots,x_n) &= f(x_1z+\sigma_1, x_2z+\sigma_2,\ldots,x_nz+\sigma_n)\\ &= \frac{\overbrace{\tilde{\alpha}_0(x_1,\ldots,x_n)\cdot z^0 + \tilde{\alpha}_1(x_1,\ldots,x_n)\cdot z + \cdots + \tilde{\alpha}_\nu(x_1,\ldots,x_n)\cdot z^\nu}^{\tilde{P}(z)\in\mathbb{K}[x_1,\ldots,x_n][z]}}{\underbrace{\tilde{\beta}_0(x_1,\ldots,x_n)\cdot z^0 + \tilde{\beta}_1(x_1,\ldots,x_n)\cdot z + \cdots + \tilde{\beta}_\delta(x_1,\ldots,x_n)\cdot z^\delta}_{\tilde{Q}(z)\in\mathbb{K}[x_1,\ldots,x_n][z]}}. \end{aligned} \tag{7}$$

In (7) terms are collected with respect to the homogenizing variable z. The coefficients $\tilde{\alpha}_0,\ldots,\tilde{\alpha}_\nu,\tilde{\beta}_0,\ldots,\tilde{\beta}_\delta$ in $\tilde{P}(z)$ and $\tilde{Q}(z)$ are multivariate polynomials in $\mathbb{K}[x_1,\ldots,x_n]$.

By the definition of the $(\sigma_1,\ldots,\sigma_n)$-shifted homogenization,

$$\begin{aligned} \tilde{Q}(z) &= \tilde{c}\cdot q(x_1z+\sigma_1,\ldots,x_nz+\sigma_n)\\ &= \tilde{c}\cdot\left(\sum_{\ell=1}^{t} b_\ell (x_1z+\sigma_1)^{e_{\ell,1}}\cdots(x_nz+\sigma_n)^{e_{\ell,n}}\right) \end{aligned}$$

for a $\tilde{c}\neq 0$. The constant term $\tilde{\beta}_0$ can be obtained by evaluating $\tilde{Q}(z)$ at 0,

$$\tilde{Q}(0) = \tilde{\beta}_0(x_1,\ldots,x_n) = \tilde{c}\cdot\sum_{\ell=1}^{t} b_\ell \sigma_1^{e_{\ell,1}}\cdots\sigma_n^{e_{\ell,n}} = \tilde{c}\cdot q(\sigma_1,\ldots,\sigma_n) \neq 0.$$

Hence we find that $\tilde{\beta}_0(x_1,\ldots,x_n)$ is a non-zero value. Note that Section 2.1 represents the special case $(\sigma_1,\ldots,\sigma_n) = (0,\ldots,0)$.

Since $\tilde{\beta}_0$ is a non-zero value, the auxiliary univariate function Γ can be normalized such that the non-zero constant in the denominator is 1,

$$\begin{aligned}\Gamma(z, x_1, \dots, x_n) &= f(x_1 z + \sigma_1, x_2 z + \sigma_2, \dots, x_n z + \sigma_n) \\ &= \frac{\overbrace{\alpha_0(x_1, \dots, x_n) + \cdots + \alpha_\nu(x_1, \dots, x_n) \cdot z^\nu}^{P(z) \in \mathbb{K}[x_1,\dots,x_n][z]}}{\underbrace{1 + \beta_1(x_1, \dots, x_n) \cdot z + \cdots + \beta_\delta(x_1, \dots, x_n) \cdot z^\delta}_{Q(z) \in \mathbb{K}[x_1,\dots,x_n][z]}}. \end{aligned} \tag{8}$$

Similar to Section 2.1, if we choose $(x_1, \dots, x_n)$ at $(\omega_1, \dots, \omega_n)$, then by a (dense) univariate rational interpolation of $\Gamma(z, \omega_1, \dots, \omega_n)$, we obtain the evaluations of $\alpha_0, \dots, \alpha_\nu, \beta_1, \dots, \beta_\delta$ at $(\omega_1, \dots, \omega_n)$ from the coefficients in

$$\begin{aligned}\Gamma(z, \omega_1, \dots, \omega_n) &= f(\omega_1 z + \sigma_1, \omega_2 z + \sigma_2, \dots, \omega_n z + \sigma_n) \\ &= \frac{\alpha_0(\omega_1, \dots, \omega_n) + \cdots + \alpha_\nu(\omega_1, \dots, \omega_n) \cdot z^\nu}{1 + \beta_1(\omega_1, \dots, \omega_n) \cdot z + \cdots + \beta_\delta(\omega_1, \dots, \omega_n) \cdot z^\delta}. \end{aligned} \tag{9}$$

But unlike Section 2.1, now the polynomial coefficients α_k and β_ℓ also collect terms due to the expansions of the shift $(\sigma_1, \dots, \sigma_n)$. So if we proceed with the simultaneous interpolation of α_k and β_ℓ from their evaluations, eventually we can recover p and q in $f = p/q$, except that such interpolation does not reflect the sparsity of the rational function in its given representation. In order to recover the sparsity, these coefficients need to be adjusted.

Note that a shift affects neither the total degrees ν and δ nor the coefficients $\alpha_\nu(x_1, \dots, x_n)$ and $\beta_\delta(x_1, \dots, x_n)$ of the highest degree terms in $P(z)$ and $Q(z)$. Moreover, only terms from the expansion of

$$c \cdot \left(\sum_{d_{k,1}+\cdots+d_{k,n}=\nu} a_k (x_1 z + \sigma_1)^{d_{k,1}} \cdots (x_n z + \sigma_n)^{d_{k,n}} \right)$$

can contribute to $\alpha_\nu(x_1, \dots, x_n)$, and similarly for $\beta_\delta(x_1, \dots, x_n)$. Hence

$$\alpha_\nu(x_1, \dots, x_n) = c \cdot \left(\sum_{d_{k,1}+\cdots+d_{k,n}=\nu} a_k x_1^{d_{k,1}} \cdots x_n^{d_{k,n}} \right),$$

$$\beta_\delta(x_1, \dots, x_n) = c \cdot \left(\sum_{e_{\ell,1}+\cdots+e_{\ell,n}=\delta} b_\ell x_1^{e_{\ell,1}} \cdots x_n^{e_{\ell,n}} \right).$$

In (6), the polynomials p and q are not required to be normalized. However, for a fixed shift $(\sigma_1, \dots, \sigma_n)$ we can require that the coefficients in p and q be normalized such that $c = 1$ and

$$\begin{aligned}\alpha_\nu(x_1, \dots, x_n) &= \sum_{d_{k,1}+\cdots+d_{k,n}=\nu} a_k x_1^{d_{k,1}} \cdots x_n^{d_{k,n}}, \\ \beta_\delta(x_1, \dots, x_n) &= \sum_{e_{\ell,1}+\cdots+e_{\ell,n}=\delta} b_\ell x_1^{e_{\ell,1}} \cdots x_n^{e_{\ell,n}}. \end{aligned} \tag{10}$$

From now on, our discussions assume the representations in (10).

We start with the (early termination) sparse interpolation of only the highest degree coefficients α_ν and β_δ. The required evaluations $\alpha_\nu(\omega_1^{(i)}, \dots, \omega_n^{(i)})$ and $\beta_\delta(\omega_1^{(i)}, \dots, \omega_n^{(i)})$ are obtained through repeated (dense) univariate rational interpolations of (9). Other evaluations of $\alpha_k(\omega_1^{(i)}, \dots, \omega_n^{(i)})$ and $\beta_\ell(\omega_1^{(i)}, \dots, \omega_n^{(i)})$ for $0 \le k < \nu$ and $1 \le \ell < \delta$ (coefficients of lower degree terms) are recorded for later interpolations.

Once both α_ν and β_δ are interpolated, we move to the next coefficients $\alpha_{\nu-1}$ and $\beta_{\delta-1}$. Note that the expansion of a shifted lower degree term can never affect the representation of higher degrees. Hence to $\alpha_{\nu-1}$, which by definition collects the coefficient of $z^{\nu-1}$ in the expansion of

$$\begin{aligned} P(z) &= \sum_{k=1}^{s} a_k (x_1 z + \sigma_1)^{d_{k,1}} \cdots (x_n z + \sigma_n)^{d_{k,n}} \\ &= \sum_{r=0}^{\nu} \left(\sum_{d_{k,1}+\cdots+d_{k,n}=r} a_k (x_1 z + \sigma_1)^{d_{k,1}} \cdots (x_n z + \sigma_n)^{d_{k,n}} \right), \end{aligned} \tag{11}$$

only (12) and (13) can contribute,

$$\sum_{d_{k,1}+\cdots+d_{k,n}=\nu} a_k(x_1z+\sigma_1)^{d_{k,1}}\cdots(x_nz+\sigma_n)^{d_{k,n}}, \tag{12}$$

$$\sum_{d_{k,1}+\cdots+d_{k,n}=\nu-1} a_k(x_1z+\sigma_1)^{d_{k,1}}\cdots(x_nz+\sigma_n)^{d_{k,n}}. \tag{13}$$

Since α_ν is already interpolated and the shift is known, the contribution to $\alpha_{\nu-1}$ from (12) can be obtained as the coefficient $u^{(\nu)}_{\nu-1}(x_1,\ldots,x_n)$ in the expansion of

$$\begin{aligned}\alpha_\nu(x_1z+\sigma_1,\ldots,x_nz+\sigma_n) &= \sum_{d_{k,1}+\cdots+d_{k,n}=\nu} a_k(x_1z+\sigma_1)^{d_{k,1}}\cdots(x_nz+\sigma_n)^{d_{k,n}}\\ &= u^{(\nu)}_\nu(x_1,\ldots,x_n)\cdot z^\nu + u^{(\nu)}_{\nu-1}(x_1,\ldots,x_n)\cdot z^{\nu-1}+\cdots+u^{(\nu)}_0(x_1,\ldots,x_n).\end{aligned}$$

We can remove the effect of the shift in the one but highest degree term accordingly. Let $\bar{\alpha}_{\nu-1}=\alpha_{\nu-1}-u^{(\nu)}_{\nu-1}$, the contribution from (13) to $\alpha_{\nu-1}$. By comparing the highest degree terms in (13), we conclude that

$$\bar{\alpha}_{\nu-1}(x_1,\ldots,x_n) = \sum_{d_{k,1}+\cdots+d_{k,n}=\nu-1} a_k x_1^{d_{k,1}}\cdots x_n^{d_{k,n}},$$

which now has a structure identical to that of $A_{\nu-1}(x_1,\ldots,x_n)$ in Section 2.1.

We proceed with the (early termination) sparse interpolation of $\bar{\alpha}_{\nu-1}$. The evaluations $\bar{\alpha}_{\nu-1}(\omega_1^{(i)},\ldots,\omega_n^{(i)})$ are computed by a similar adjustment of the stored $\alpha_{\nu-1}(\omega_1^{(i)},\ldots,\omega_n^{(i)})$,

$$\bar{\alpha}_{\nu-1}(\omega_1^{(i)},\ldots,\omega_n^{(i)}) = \alpha_{\nu-1}(\omega_1^{(i)},\ldots,\omega_n^{(i)}) - u^{(\nu)}_{\nu-1}(\omega_1^{(i)},\ldots,\omega_n^{(i)}).$$

At this stage, the number of the stored $\alpha_{\nu-1}(\omega_1^{(i)},\ldots,\omega_n^{(i)})$ is sufficient for the earlier interpolations of $\bar{\alpha}_\nu$ and $\bar{\beta}_\delta$, the coefficients of the higher degree terms. If the number of stored evaluations $\alpha_{\nu-1}(\omega_1^{(i)},\ldots,\omega_n^{(i)})$ does not produce enough $\bar{\alpha}_{\nu-1}(\omega_1^{(i)},\ldots,\omega_n^{(i)})$ for the current interpolation of $\bar{\alpha}_{\nu-1}$, more evaluations of $\alpha_{\nu-1}$ can be added through new univariate rational interpolations of $\Gamma(z,\omega_1^{(i)},\ldots,\omega_n^{(i)})$ at additional i as in (9).

Instead of interpolating the $A_0,\ldots,A_\nu$ in parallel as in Section 2.1, our shifting strategy interpolates the adjusted $\bar{\alpha}_\nu,\bar{\alpha}_{\nu-1},\ldots,\bar{\alpha}_0$ sequentially. We start from the highest degree term $\bar{\alpha}_\nu=\alpha_\nu$ and interpolate $\bar{\alpha}_\nu$. For $r=0,\ldots,\nu-1$, after interpolating

$$\bar{\alpha}_{\nu-r}(x_1,\ldots,x_n) = \sum_{d_{k,1}+\cdots+d_{k,n}=\nu-r} a_k x_1^{d_{k,1}}\cdots x_n^{d_{k,n}},$$

the $u_0^{(\nu-r)},u_1^{(\nu-r)},\ldots,u_{\nu-r-1}^{(\nu-r)}$ are computed from the expansion

$$\begin{aligned}\bar{\alpha}_{\nu-r}(x_1z+\sigma_1,\ldots,x_nz+\sigma_n) &= \sum_{d_{k,1}+\cdots+d_{k,n}=\nu-r} a_k(x_1z+\sigma_n)^{d_{k,1}}\cdots(x_nz+\sigma_n)^{d_{k,n}}\\ &= u^{(\nu-r)}_{\nu-r}(x_1,\ldots,x_n)\cdot z^{\nu-r} + u^{(\nu-r)}_{\nu-r-1}(x_1,\ldots,x_n)\cdot z^{\nu-r-1}+\cdots+u^{(\nu-r)}_0(x_1,\ldots,x_n).\end{aligned}$$

For $r=1,\ldots,\nu$, the adjusted $\bar{\alpha}_{\nu-r}$ is obtained by removing from $\alpha_{\nu-r}$ the contribution denoted by $U_{\nu-r}$,

$$\bar{\alpha}_{\nu-r} = \alpha_{\nu-r} - \underbrace{\sum_{j=0,\ldots,r-1} u^{(\nu-j)}_{\nu-r}}_{U_{\nu-r}}. \tag{14}$$

From the evaluations of (14), $\bar{\alpha}_{\nu-r}$ can be interpolated and then included in the adjustment for the next $\bar{\alpha}_{\nu-r-1}$. We interpolate every newly adjusted $\bar{\alpha}_{\nu-r}$ until all $\bar{\alpha}_k$ are interpolated. Then

$$p=\sum_{r=0}^{\nu}\bar{\alpha}_{\nu-r}$$

is known.

The interpolation of q can be carried out in parallel. We start with $\bar{\beta}_\delta=\beta_\delta$ and interpolate $\bar{\beta}_\delta$. For $r=0,\ldots,\delta-1$, each time $\bar{\beta}_{\delta-r}(x_1,\ldots,x_n)$ is interpolated, the $v_0^{(\delta-r)},v_1^{(\delta-r)},\ldots,v_{\delta-r-1}^{(\delta-r)}$ are computed from the expansion of $\bar{\beta}_{\delta-r}(x_1z+\sigma_1,\ldots,x_nz+\sigma_n)$,

$$\sum_{e_{\ell,1}+\cdots+e_{\ell,n}=\delta-r} b_\ell(x_1z+\sigma_n)^{e_{\ell,1}}\cdots(x_nz+\sigma_n)^{e_{\ell,n}} = v^{(\delta-r)}_{\delta-r}(x_1,\ldots,x_n)\cdot z^{\delta-r}+\cdots+v^{(\delta-r)}_0(x_1,\ldots,x_n).$$

For $r = 1, \ldots, \delta$, removing from $\beta_{\delta-r}$ the contribution denoted by $V_{\delta-r}$ gives

$$\bar{\beta}_{\delta-r} = \beta_{\delta-r} - \underbrace{\sum_{j=0,\ldots,r-1} v_{\delta-r}^{(\delta-j)}}_{V_{\delta-r}}. \tag{15}$$

From the evaluations of (15), $\bar{\beta}_{\delta-r}$ can be interpolated and included in the adjustment for the next $\bar{\beta}_{\delta-r-1}$. When all $\bar{\beta}_\ell$ are interpolated, then

$$q = \sum_{r=0}^{\delta} \bar{\beta}_{\delta-r}$$

is known.

An attractive feature of the sparse polynomial interpolation of the adjusted $\bar{\alpha}_k$ and $\bar{\beta}_\ell$ is that it reflects the original sparsity of p and q in $f = p/q$. Hence our overall interpolation method is sensitive to the sparsity of the rational function in its originally given multinomial representation.

Algorithm: `Sparse Rational Interpolation <general>`

Input:
- $f(x_1, \ldots, x_n)$: a multivariate black box rational function.
- ν and δ: total degrees of the numerator p and denominator q.
- η: a positive integer (or default to 1), the threshold required by the early termination strategy [16].

Output:
- $a_k, b_\ell, (d_{k,1}, \ldots, d_{k,n}), (e_{\ell,1}, \ldots, e_{\ell,n})$:

$$f(x_1, \ldots, x_n) = \frac{\sum_{k=1}^{s} a_k x_1^{d_{k,1}} \cdots x_n^{d_{k,n}}}{\sum_{\ell=1}^{t} b_\ell x_1^{e_{\ell,1}} \cdots x_n^{e_{\ell,n}}} \text{ with high probability,}$$

or an error message if the procedure fails to complete [16].

Steps:

1. [Shift.]
   If $f(\sigma_1, \ldots, \sigma_n)$ is defined, $(\sigma_1, \ldots, \sigma_n)$ can be chosen as a shift.
2. [Shifted homogenization.]
   For $i = 0, 1, 2, \ldots$:
   Generate $(\omega_1^{(i)}, \ldots, \omega_n^{(i)})$ as dictated by the chosen sparse polynomial interpolation algorithm.
3. [Dense univariate rational interpolation.]
   Pick distinct $\zeta_0, \zeta_1, \ldots, \zeta_{\nu+\delta}$ and evaluate $f(\omega_1^{(i)}\zeta_j + \sigma_1, \ldots, \omega_n^{(i)}\zeta_j + \sigma_n)$ for $0 \le j \le \nu + \delta$. From the $\nu + \delta + 1$ evaluations interpolate

$$\begin{aligned} \Gamma(z, \omega_1^{(i)}, \ldots, \omega_n^{(i)}) &= f(\omega_1^{(i)} z + \sigma_1, \ldots, \omega_n^{(i)} z + \sigma_n) \\ &= \frac{\alpha_0(\omega_1^{(i)}, \ldots, \omega_n^{(i)}) \cdot z^0 + \cdots + \alpha_\nu(\omega_1^{(i)}, \ldots, \omega_n^{(i)}) \cdot z^\nu}{1 + \beta_1(\omega_1^{(i)}, \ldots, \omega_n^{(i)}) \cdot z + \cdots + \beta_\delta(\omega_1^{(i)}, \ldots, \omega_n^{(i)}) \cdot z^\delta}. \end{aligned} \tag{16}$$

   The values of $\alpha_{\nu-1}(\omega_1^{(i)}, \ldots, \omega_n^{(i)}), \ldots, \alpha_0(\omega_1^{(i)}, \ldots, \omega_n^{(i)}), \beta_{\delta-1}(\omega_1^{(i)}, \ldots, \omega_n^{(i)}), \ldots, \beta_1(\omega_1^{(i)}, \ldots, \omega_n^{(i)})$ are stored for later interpolations.
4. [Initialize and interpolate $\bar{\alpha}_\nu$ and $\bar{\beta}_\delta$.]
   Use the early termination sparse interpolation to continue the interpolations of $\bar{\alpha}_\nu = \alpha_\nu$ and $\bar{\beta}_\delta = \beta_\delta$ from evaluations of $\alpha_\nu(\omega_1^{(i)}, \ldots, \omega_n^{(i)})$ and $\beta_\delta(\omega_1^{(i)}, \ldots, \omega_n^{(i)})$.
   If both $\bar{\alpha}_\nu$ and $\bar{\beta}_\delta$ are interpolated, then break out of the i loop.
5. [Update $\bar{\alpha}_{\nu-r}$ and $\bar{\beta}_{\delta-r}$.]
   Initialize $U_{\nu-1} = U_{\nu-2} = \cdots = U_0 = 0$ and $V_{\delta-1} = V_{\delta-2} = \cdots = V_1 = 0$.
   For $r = 1, \ldots, \max(\nu, \delta)$:
   At this stage both $\bar{\alpha}_{\nu+1-r}$ and $\bar{\beta}_{\delta+1-r}$ are interpolated and we have

$$\bar{\alpha}_{\nu+1-r}(x_1 z + \sigma_1, \ldots, x_n z + \sigma_n) = \sum_{j=0}^{\nu+1-r} u_j^{(\nu+1-r)}(x_1, \ldots, x_n) \cdot z^j,$$

$$\bar{\beta}_{\delta+1-r}(x_1 z + \sigma_1, \ldots, x_n z + \sigma_n) = \sum_{j=0}^{\delta+1-r} v_j^{(\delta+1-r)}(x_1, \ldots, x_n) \cdot z^j.$$

For $j = r, \ldots, \max(\nu, \delta - 1)$:

$$U_{\nu-j} \longleftarrow U_{\nu-j} + u_{\nu-j}^{(\nu+1-r)} \quad \text{and} \quad V_{\delta-j} \longleftarrow V_{\delta-j} + v_{\delta-j}^{(\delta+1-r)}$$

End of j.

6. [Interpolate $\bar{\alpha}_{\nu-r}$ and $\bar{\beta}_{\delta-r}$.]

Use the early termination sparse interpolation to interpolate $\bar{\alpha}_{\nu-r}$ and $\bar{\beta}_{\delta-r}$ from the evaluations

$$\bar{\alpha}_{\nu-r}(\omega_1^{(i)}, \ldots, \omega_n^{(i)}) = \alpha_{\nu-r}(\omega_1^{(i)}, \ldots, \omega_n^{(i)}) - U_{\nu-r}(\omega_1^{(i)}, \ldots, \omega_n^{(i)}),$$
$$\bar{\beta}_{\delta-r}(\omega_1^{(i)}, \ldots, \omega_n^{(i)}) = \beta_{\delta-r}(\omega_1^{(i)}, \ldots, \omega_n^{(i)}) - V_{\delta-r}(\omega_1^{(i)}, \ldots, \omega_n^{(i)}).$$

If both $\bar{\alpha}_{\nu-r}$ and $\bar{\beta}_{\delta-r}$ are interpolated, then increase r.

If the stored values $\alpha_{\nu-r}(\omega_1^{(i)}, \ldots, \omega_n^{(i)})$ or $\beta_{\delta-r}(\omega_1^{(i)}, \ldots, \omega_n^{(i)})$ are not sufficient for interpolating either $\bar{\alpha}_{\nu-r}$ or $\bar{\beta}_{\delta-r}$, then

continue with the i loop stopped earlier for continuing the interpolation of $\bar{\alpha}_{\nu-r}$ and/or $\bar{\beta}_{\delta-r}$. If both $\bar{\alpha}_{\nu-r}$ and $\bar{\beta}_{\delta-r}$ are interpolated, then break out of the i loop and increase r.

End of r.

The number of required black box evaluations is similar to Section 2.1. Each interpolation of $\Gamma(z, \omega_1^{(i)}, \ldots, \omega_n^{(i)})$ needs $O(\nu + \delta)$ evaluations. We perform a sparse interpolation of $\bar{\alpha}_k$ and $\bar{\beta}_\ell$ (comparable to A_k and B_ℓ in Section 2.1) of which the number of terms is bounded by the sparsity of p and q. Hence the overall i loop is bounded by the sparsity of either p or q in $f = p/q$ with respect to its given representation. Adjusting α_k to $\bar{\alpha}_k$ and β_ℓ to $\bar{\beta}_\ell$ may introduce additional operations such as the polynomial expansions in Step 5 and the evaluations of $U_{\nu-r}$ and $V_{\delta-r}$ in Step 6, but does not require additional evaluations of the multivariate rational function f.

3. Implementations and illustrated examples

The combination of different (dense) algorithms for univariate rational interpolation and sparse polynomial interpolation leads to a class of sparse multivariate rational interpolation algorithms.

In Section 3.1, we discuss implementations in exact arithmetic. Section 3.2 reports on our approach in floating point arithmetic. The effectiveness and practicality of our method are illustrated in several examples.

3.1. Exact arithmetic

Let $\mathbb{K}$ be a field. We comment on the combination of (dense) univariate rational interpolation and sparse polynomial interpolation algorithms in our method.

3.1.1. Dense univariate rational interpolation

Our auxiliary univariate functions are interpolated densely. The classical problem of univariate rational interpolation determines the coefficients α_k and β_ℓ in

$$\Gamma(z) = \frac{\overbrace{\alpha_0 + \alpha_1 z + \cdots + \alpha_\nu z^\nu}^{P(z)}}{\underbrace{1 + \beta_1 z + \cdots + \beta_\delta z^\delta}_{Q(z)}} \in \mathbb{K}(z)$$

from the evaluations $\Gamma(\zeta_0), \Gamma(\zeta_1), \ldots, \Gamma(\zeta_{\nu+\delta})$. Suppose $\deg(P) = \nu$ and $Q(z) = \delta$ are given and that all the evaluations of $\Gamma(\zeta_j)$ are defined, which implies $Q(\zeta_j) \neq 0$ for $0 \leq j \leq \nu + \delta$. It is well-known that the interpolation problem can be solved from the linear system of equations,

$$\Gamma(\zeta_j)Q(\zeta_j) - P(\zeta_j) = 0, \quad j = 0, 1, \ldots, \nu + \delta. \tag{17}$$

On the other hand, by the Chinese Remainder Theorem, the above problem can also be formulated as

$$P(z) \equiv Q(z)\Gamma(z), \quad \bmod (z - \zeta_0)(z - \zeta_1) \cdots (z - \zeta_{\nu+\delta}) \tag{18}$$

in which $P(z)$ and $Q(z)$ can be computed by the extended Euclidean algorithm [25]. As for recovering the coefficients in $P(z)$ and $Q(z)$ when $\mathbb{K} = \mathbb{Z}_\rho$ and ρ is a prime, the fast Euclidean algorithm can be modified for fast rational function reconstruction [21]. When $\mathbb{K} = \mathbb{Q}$, modular arithmetic can be combined with rational vector recovery to simultaneously recover the common denominator of rational coefficients [22,19,23].

3.1.2. Sparse multivariate polynomial interpolation

The coefficients of each auxiliary univariate rational function require sparse multivariate polynomial interpolation algorithms.

Without the sparsity supplied as input, we can employ the early termination Ben-Or/Tiwari algorithm [17,16] or Zippel's interpolation [26]. Both algorithms are sensitive to the sparsity of the target polynomial and deliver results that are correct with high probability.

As for interpolating over a small finite field, without an extension of the ground field an exponential lower bound $\Omega(n^{\log\tau})$ where $\tau = \max(s, t)$ for the number of black box queries is established in [2]. In [12] an efficient algorithm is obtained through a slight field extension. On the other hand, [16, Section 5.2] shows that without field extension a sparse multivariate polynomial can be interpolated from $O(n\tau)$ black box evaluations with high probability. With respect to the exponential lower bound for multivariate sparse interpolation given in [2], this should be interpreted as performing probabilistic univariate sparse interpolation in a variable-by-variable manner. Similarly, with high probability, the parallel algorithm in [14] also achieves sparse interpolation from $O(n\tau)$ evaluations because it uses variable-by-variable evaluations in a bipartite graph matching.

A sparse algorithm is less efficient when interpolating a dense polynomial. Hence, the racing algorithm [17,16] is developed to run a dense against a sparse interpolation on a same set of evaluations. In general, the overall racing algorithm is superior because it terminates as soon as any of the racer algorithms terminates and requires no additional evaluations in comparison to a single algorithm.

3.1.3. An exact example in finite field arithmetic

Consider the reconstruction of

$$f(x_1, x_2, x_3) = \frac{p(x_1, x_2, x_3)}{q(x_1, x_2, x_3)} = \frac{x_1^4 + 3x_2^5 + x_3^2}{2x_1x_2x_3^2 + 3x_2} \mod 3137 \tag{19}$$

with its total degrees $\deg(p) = 5$, $\deg(q) = 4$ given. In general, there are 56 possible terms in the numerator and 35 in the denominator. Even with normalization, classical rational interpolation requires at least $56 + 35 - 1 = 90$ evaluations to recover $f(x_1, x_2, x_3)$.

We illustrate our sparse method. The function $f(x_1, x_2, x_3)$ is not defined at $(0, 0, 0)$, so we pick a shift, say $(2, 1, 1)$ and consider the interpolation of $\Gamma(x_1z + 2, x_2z + 1, x_3z + 1)$. We evaluate (x_1, x_2, x_3) at $(2^i, 3^i, 5^i)$ for $i = 1, 2, \ldots$. For each i, we interpolate $\Gamma(2^iz + 2, 3^iz + 1, 5^iz + 1)$ at $z = 1, \ldots, 10$. Our interpolation result agrees with f in $\mathbb{Z}_{3137}$,

$$\frac{2689x_1^4 + 1793x_2^5 + 2689x_3^2}{2241x_1x_2x_3^2 + 1793x_2} \equiv \frac{x_1^4 + 3x_2^5 + x_3^2}{2x_1x_2x_3^2 + 3x_2} = f(x_1, x_2, x_3) \mod 3137.$$

Each $\Gamma(2^iz+2, 3^iz+1, 5^iz+1)$ is interpolated from 10 black box evaluations. Using the early termination Ben-Or/Tiwari algorithm, with high probability, each adjusted coefficient in Γ can be interpolated after 3 evaluations. The multivariate rational function f in (19) can thus be interpolated from 30 black box evaluations in total.

A fine analysis shows that this number of 30 probes can be brought down a bit more. By normalizing the denominator's non-zero constant to 1, the interpolation of $\Gamma(2z + 2, 3z + 1, 5z + 1)$ requires $(\nu + \delta + 1)$ evaluations. Note that the constant in the numerator remains fixed for all $\Gamma(2^iz + 2, 3^iz + 1, 5^iz + 1)$. Once $\Gamma(2z + 2, 3z + 1, 5z + 1)$ is interpolated, the numerator's constant is also known and each interpolation of $\Gamma(2^iz + 2, 3^iz + 1, 5^iz + 1)$ can be achieved from $\nu + \delta$, instead of $\nu+\delta+1$, evaluations. So our example can actually be interpolated after $10+9+9 = 28$ black box evaluations of f.

3.1.4. Comparison and remarks

The overall performance of our rational interpolation reflects the choice of the sparse interpolation algorithms employed, e.g., the early termination version of the Ben-Or/Tiwari algorithm, Zippel's algorithm, or the racing algorithm that races Ben-Or/Tiwari against Zippel [16].

We compare our method with the exact multivariate sparse rational interpolation in [19]. Both are probabilistic algorithms, which means the results are correct with high probability. Both use the racing algorithm that races Ben-Or/Tiwari against Zippel [16]. If we use the early termination version of the Ben-Or/Tiwari algorithm with threshold η, our algorithm requires no more than $(\nu+\delta+1)+(2\max(s, t) + \eta)(\nu+\delta)$ black box probes (see the last line in Section 3.1.3 to understand this count). To our knowledge this yields an efficient algorithm that at this moment best reflects and exploits the sparsity of the rational function in terms of the number of evaluations.

Fig. 1 reports some comparisons on the number of black box probes required by the Sparse Rational Function Interpolation (KY) in [19] and our method employing the racing algorithm in [16] (CL).

As shown in Fig. 1, in some cases our method requires only a fraction of the number of black box probes required by [19]. This is because our method does not directly interpolate the numerator and denominator polynomials but rather the coefficients in the univariate auxiliary function. These coefficients are possibly even sparser polynomials, which is a gain obtained for free from performing a dense rational interpolation in the homogenizing variable z.

3.2. Floating point arithmetic

Several numerical challenges are encountered when attempting to implement a sparse interpolation method in floating point arithmetic. That is, when both the inputs and outputs of the black box rational function have some errors, and all numbers are represented in fixed precision arithmetic.

Ex.	Coeff. Range	ν, δ	n	s, t	mod	KY	CL
1	[−10, 10]	(3, 3)	2	6, 6	503	221	85
2	[−10, 10]	(5, 2)	4	6, 3	1009	339	211
3	[−20, 20]	(2, 4)	6	2, 5	120011	357	133
4	[−20, 20]	(1, 6)	8	4, 8	8009	777	659
5	[−30, 30]	(10, 5)	10	7, 4	4001	2246	1471
6	[−10, 10]	(15, 15)	15	15, 15	50021	17120	8191
7	[−10, 10]	(20, 20)	20	20, 20	50021	38682	18281
8	[−30, 30]	(30, 15)	5	20, 10	10007	12896	2071
10	[−50, 50]	(50, 50)	50	50, 50	1000003	603638	330901
11	[−10, 10]	(2, 8)	90	10, 50	1000003	75082	61791

Fig. 1. Comparing the number of black box evaluations on benchmarks from [19].

With or without a shift, our method always normalizes a non-zero constant in the denominator. But in a numerical setting, a non-zero constant may be very small hence can be regarded as numerically zero. Normalizing such a small constant value can introduce ill-conditioning. Moreover, an introduced shift is powered along the degree of each monomial in both the numerator and denominator, which can also skew the scale of the original function.

We present an approach that allows evaluations and computations over the complex field and explores a random shift on the unit circle. Our approach leads to a heuristic for stable normalization that preserves the scale of the original rational function.

3.2.1. Stable normalization in the univariate rational interpolation

If the given rational function is defined at $(0, \ldots, 0)$, we follow Section 2.1 and proceed with the interpolation by normalizing the non-zero constant in the denominator. If not, Section 2.2 shows that a non-zero constant can be enforced through a shift $(\sigma_1, \ldots, \sigma_n)$, where the given function is defined.

In a finite precision environment, the constant in the denominator can be very small, even numerically regarded as zero. If we proceed with normalizing a small or numerically zero constant, we may lose the sparsity by treating a numerically zero term as non-zero. Moreover, numerical instabilities, due to scaling a very small value (zero) to one, can be introduced at the same time.

However, there exist shifts that can lead to a numerically non-zero constant for a stable normalization. Recall that in Section 2.2 we introduce the $(\sigma_1, \ldots, \sigma_n)$-shifted homogenization $\Gamma(z, x_1, \ldots, x_n)$ of the given multivariate rational function $f = p/q$,

$$\begin{aligned}\Gamma(z, x_1, \ldots, x_n) &= f(x_1 z + \sigma_1, \ldots, x_n z + \sigma_n) = \frac{p(x_1 z + \sigma_1, \ldots, x_n z + \sigma_n)}{q(x_1 z + \sigma_1, \ldots, x_n z + \sigma_n)} \\ &= \frac{\tilde{\alpha}_0(x_1, \ldots, x_n) + \tilde{\alpha}_1(x_1, \ldots, x_n) \cdot z + \cdots + \tilde{\alpha}_\nu(x_1, \ldots, x_n) \cdot z^\nu}{\tilde{\beta}_0(x_1, \ldots, x_n) + \tilde{\beta}_1(x_1, \ldots, x_n) \cdot z + \cdots + \tilde{\beta}_\delta(x_1, \ldots, x_n) \cdot z^\delta},\end{aligned}$$

and that $\tilde{\beta}_0(x_1, \ldots, x_n) = c \cdot q(\sigma_1, \ldots, \sigma_n)$ is a constant value.

Now let $f = \hat{p}/\hat{q}$ with the polynomials $\hat{p}, \hat{q} \in \mathbb{Z}[\mathtt{i}][x_1, \ldots, x_n]$. In other words, the coefficients of $\hat{p}$, $\hat{q}$ are all on the integer lattice of the complex plane, which can always be achieved in a finite precision environment. The value of $\hat{q}(\sigma_1, \ldots, \sigma_n)$ corresponds to the constant in the denominator of $\Gamma(z, x_1, \ldots, x_n)$. By a numeric Zippel–Schwartz lemma [20, Lemma 3.1], at certain random points, such evaluation can be expected to be bounded away from 0 guaranteeing a numerically non-zero constant. In [20], this numerical Zippel–Schwartz lemma is used as a (partially) mathematical justification for the heuristics of deciding whether a polynomial is identically zero from its evaluations.

Lemma (*Numerical Zippel–Schwartz [20, Lemma 3.1]*). *Let $0 \not\equiv \hat{q}(x_1, \ldots, x_n) \in \mathbb{Z}[\mathtt{i}][x_1, \ldots, x_n]$ and for $1 \le j \le n$ let $\sigma_j = \exp(2\pi\mathtt{i}/\rho_j) \in \mathbb{C}$ with $\rho_j \in \mathbb{Z}_{\ge 3}$ distinct prime numbers. Suppose $\hat{q}(\sigma_1, \ldots, \sigma_n) \ne 0$. Then for random integers θ_j with $1 \le \theta_j < \rho_j$ the expected value*

$$E\{|\hat{q}(\sigma_1^{\theta_1}, \ldots, \sigma_n^{\theta_n})|\} \ge 1.$$

By the Zippel–Schwartz lemma [6,26,24], through randomization, the premise of $\hat{q}(\sigma_1, \ldots, \sigma_n) \neq 0$ can be obtained with high probability. The lemma asserts, with high probability, a non-zero constant $\hat{q}(\sigma_1^{\theta_1}, \ldots, \sigma_n^{\theta_n})$ in the $(\sigma_1^{\theta_1}, \ldots, \sigma_n^{\theta_n})$-shifted homogenization, in which both $(\sigma_1, \ldots, \sigma_n)$ and $(\theta_1, \ldots, \theta_n)$ are randomized.

Once a (stable) normalization is achieved, we can proceed with the interpolation of the auxiliary univariate rational function as in (17). This problem has been studied extensively in a numerical setting.

3.2.2. Numerical sparse multivariate polynomial interpolation

After a set of auxiliary univariate rational functions are interpolated, we proceed with the interpolation of their coefficients that are multivariate polynomials.

In floating point arithmetic, without the knowledge of the sparsity, an iterative algorithm based on the qd-scheme, can interpolate all the variables at once. It is still sensitive to the sparsity of the target polynomial [4] which it detects while running. A heuristic analogue to Zippel's variable-by-variable sparse algorithm can also recover polynomials from noisy evaluations [20].

The numerical sparse polynomial interpolation algorithms in [8,9] require the sparsity to be supplied as input. But it was noticed in [9, Section 4.7] that the sparsity can be reflected in the conditioning of the associated Hankel systems and all their leading minors, which leads to a heuristic of probabilistically detecting sparsity in a numerical setting.

3.2.3. An example with noisy evaluations

In floating point arithmetic, we consider the same black box rational function as in (19)

$$f(x_1, x_2, x_3) = \frac{p(x_1, x_2, x_3)}{q(x_1, x_2, x_3)} = \frac{x_1^4 + 3x_2^5 + x_3^2}{2x_1x_2x_3^2 + 3x_2},$$

with the total degrees $\deg(p) = \nu = 5$ and $\deg(q) = \delta = 4$ given.

We test our method in Maple. We set `Digits` to 15 and add random noise between 10^{-3} and 10^{-5} to each black box evaluation of f.

The function $f(x_1, x_2, x_3)$ is not defined at the origin. A shift $(\sigma_1, \sigma_2, \sigma_3)$ is picked at which f is defined

$$\sigma_1 = \exp(2\pi\,\mathtt{i}/3), \qquad \sigma_2 = \exp(4\pi\,\mathtt{i}/3), \qquad \sigma_3 = \exp(4\pi\,\mathtt{i}/3).$$

Now the normalization of the $(\sigma_1, \sigma_2, \sigma_3)$-shifted homogenization Γ can be achieved. For each i, we interpolate

$$\Gamma(\exp(2\pi\,\mathtt{i}/5)^i z + \sigma_1, \exp(2\pi\,\mathtt{i}/7)^i z + \sigma_2, \exp(2\pi\,\mathtt{i}/3)^i z + \sigma_3)$$

from the evaluations at $z = \exp(2j\pi\,\mathtt{i}/10)$ for $j = 1, \dots, 10$.

Using the numerical sparse polynomial interpolation in [8,9] we determine each adjusted coefficient in Γ after 4 evaluations. Let $f = p/q = \bar{p}/\bar{q}$ such that $\bar{p}$, $\bar{q}$ are polynomials normalized with respect to the non-zero constant in the denominator of $\Gamma(x_1z + \sigma_1, x_2z + \sigma_2, x_3z + \sigma_3)$. After 40 black box evaluations we obtain an interpolant $\tilde{f} = \tilde{p}/\tilde{q}$ with a relative error of the magnitude of the noise:

$$\frac{\|\tilde{p} - \bar{p}\|_2^2 + \|\tilde{q} - \bar{q}\|_2^2}{\|\bar{p}\|_2^2 + \|\bar{q}\|_2^2} = 0.00131\dots.$$

Note that in order to correctly recover x_1, x_2, x_3 at once, we need enough precision to distinguish powers of $\exp(2\pi\,\mathtt{i}/m)$ where $m = 5 \times 7 \times 3$. Clearly, higher degrees of the primitive roots of unity require a higher precision in the computations. In the multivariate case, such problem can be traded off for interpolating fewer variables at a time. We also mention that in [9] it is shown that the conditioning can be improved through randomization.

3.2.4. Comparison and future research

The numerical behavior of our overall interpolation method depends on the performance of the interpolation algorithms employed for univariate rational functions and multivariate polynomials. Sections 3.2.1 and 3.2.2 show that it is possible to achieve reasonable numerical robustness through randomization over complex values. We demonstrate some prospects of our approach through examples and describe some key issues under investigation.

Our method interpolates the numerical example in Section 3.2.3 from 40 black box probes. The numerical algorithm ZNIPR [20] interpolates the same example after 98 black box probes. ZNIPR can use oversampling to achieve better conditioning. But for comparison we set the oversampling for this example to 0 and the noise range somewhat smaller between 10^{-4} and 10^{-6}. Then the returned interpolant is in the same error range.

We further test ZNIPR without oversampling against our approach, also without oversampling, on the first three examples in Table 2 of [20, Section 4.3]. Employing the sparse polynomial interpolation in [9], our approach delivers comparable results in fewer black box probes. All the relative errors of the interpolants are of the same magnitude as that of the noise, which is as good as one can expect (see Fig. 2).

As indicated earlier, the conditioning of the interpolation problem can be improved through randomization. This is currently under investigation. In addition, a probability analysis of the shift strategy needs to be performed as well.

To conclude, we point out that in many applications one seeks to fit high dimensional data with a (preferably sparse) rational model. As the number of possible terms explodes exponentially in the number of variables n, the possible size of the interpolation problem can be huge even when the total degree is not high. Our multivariate sparse method seems to be very suitable for such a scenario. We are therefore planning to explore different and specific applications.

				ZNIPR		Our approach	
Ex.	ν, δ	s, t	n	random noise	probes	random noise	probes
1	1,1	2,2	2	$10^{-5} \sim 10^{-3}$	28	$10^{-5} \sim 10^{-3}$	12
2	2,2	3,3	2	$10^{-4} \sim 10^{-6}$	40	$10^{-5} \sim 10^{-3}$	25
3	1,4	2,4	3	$10^{-4} \sim 10^{-6}$	73	$10^{-6} \sim 10^{-8}$	60

Fig. 2. Some numerical tests for sparse multivariate rational interpolation.

Acknowledgements

We thank Erich Kaltofen and Zhengfeng Yang for valuable remarks and providing their sparse rational interpolation codes and benchmarks. We acknowledge the anonymous referees for their constructive comments.

References

[1] H. Allouche, A. Cuyt, On the structure of a table of multivariate rational interpolants, Constr. Approx. 8 (1992) 69–86.
[2] M. Clausen, A. Dress, J. Grabmeier, M. Karpinski, On zero-testing and interpolation of k-sparse multivariate polynomials over finite fields, Theoret. Comput. Sci. 84 (2) (1991) 151–164.
[3] A. Cuyt, A recursive computation scheme for multivariate rational interpolants, SIAM J. Numer. Anal. 24 (1987) 228–238.
[4] A. Cuyt, W.-s. Lee, A new algorithm for sparse interpolation of multivariate polynomials, Theoret. Comput. Sci. 409 (2) (2008) 180–185.
[5] J. de Kleine, M. Monagan, A. Wittkopf, Algorithms for the non-monic case of the sparse modular gcd algorithm, in: X.-S. Gao, G. Labahn (Eds.), ISSAC '05: Proceedings of the 2005 International Symposium on Symbolic and Algebraic Computation, ACM Press, 2005, pp. 124–131.
[6] R.A. DeMillo, R.J. Lipton, A probabilistic remark on algebraic program testing, Inform. Process. Lett. 7 (4) (1978) 193–195.
[7] A. Díaz, E. Kaltofen, FoxBox: a system for manipulating symbolic objects in black box representation, in: O. Gloor (Ed.), Proc. 1998 Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation, ISSAC'98, ACM Press, 1998, pp. 30–37.
[8] M. Giesbrecht, G. Labahn, W.-s. Lee, Symbolic-numeric sparse interpolation of multivariate polynomials (extended abstract), in: J.-G. Dumas (Ed.), ISSAC MMVI Proc. 2006 Internat. Symp. Symbolic Algebraic Comput., ACM Press, 2006, pp. 116–123.
[9] M. Giesbrecht, G. Labahn, W.-s. Lee, Symbolic-numeric sparse interpolation of multivariate polynomials, J. Symbolic Comput. 44 (8) (2009) 943–959.
[10] D. Grigoriev, M. Karpinski, M.F. Singer, Computational complexity of sparse rational interpolation, SIAM J. Comput. 23 (1) (1994) 1–12.
[11] D.Y. Grigoriev, M. Karpinski, Algorithms for sparse rational interpolation, in: S.M. Watt (Ed.), ISSAC '91: Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation, ACM Press, 1991, pp. 7–13.
[12] D.Y. Grigoriev, M. Karpinski, M.F. Singer, Fast parallel algorithms for sparse multivariate polynomial interpolation over finite fields, SIAM J. Comput. 19 (6) (1990) 1059–1063.
[13] D.Y. Grigoriev, M. Karpinski, M.F. Singer, Interpolation of sparse rational functions without knowing bounds on exponents, in: FOCS 1990: Proceedings of the 31st Annual Symposium on Foundations of Computer Science, vol.2, IEEE Computer Society, 1990, pp. 840–846.
[14] S.M.M. Javadi, M. Monagan, Parallel sparse polynomial interpolation over finite fields, in: Proc. of Parallel Symbolic Computation, PASCO, 2010, ACM Press, New York, N.Y, 2010, pp. 160–168.
[15] E. Kaltofen, Greatest common divisors of polynomials given by straight-line programs, J. ACM 35 (1) (1988) 231–264.
[16] E. Kaltofen, W.-s. Lee, Early termination in sparse interpolation algorithms, J. Symbolic Comput. 36 (3–4) (2003) 365–400.
[17] E. Kaltofen, W.-s. Lee, A.A. Lobo, Early termination in Ben-Or/Tiwari sparse interpolation and a hybrid of Zippel's algorithm, in: C. Traverso (Ed.), Proc. 2000 Interna. Symp. Symbolic Algebraic Comput., ISSAC'00, ACM Press, 2000, pp. 192–201.
[18] E. Kaltofen, B. Trager, 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.
[19] E. Kaltofen, Z. Yang, On exact and approximate interpolation of sparse rational functions, in: C.W. Brown (Ed.), ISSAC'07: Proceedings of the 2007 International Symposium on Symbolic and Algebraic Computation, ACM, 2007, pp. 203–210.
[20] E. Kaltofen, Z. Yang, L. Zhi, On probabilistic analysis of randomization in hybrid symbolic-numeric algorithms, in: J. Verschelde, S.M. Watt (Eds.), SNC '07: Proceedings of the 2007 International Workshop on Symbolic-Numeric Computation, ACM, 2007, pp. 11–17.
[21] S. Khodadad, M. Monagan, Fast rational function reconstruction, in: J.-G. Dumas (Ed.), ISSAC MMVI Proc. 2006 Internat. Symp. Symbolic Algebraic Comput., ACM Press, 2006, pp. 184–190.
[22] M. Monagan, Maximal quotient rational reconstruction: an almost optimal algorithm for rational reconstruction, in: J. Gutierrez (Ed.), ISSAC '04: Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation, ACM Press, 2004, pp. 243–249.
[23] Z. Olesh, A. Storjohann, The vector rational function reconstruction problem, in: Proceedings of the Waterloo Workshop: Computer Algebra 2006, Latest Advances in Symbolic Algorithms, World Scientific, 2007, pp. 137–149.
[24] J.T. Schwartz, Fast probabilistic algorithms for verification of polynomial identities, J. ACM 27 (1980) 701–717.
[25] J. von zur Gathen, J. Gerhard, Modern Computer Algebra, Cambridge University Press, 1999.
[26] R. Zippel, Probabilistic algorithms for sparse polynomials, in: Proc. EUROSAM '79, in: Lect. Notes Comput. Sci., vol. 72, Springer Verlag, 1979, pp. 216–226.