

THE VECTOR RATIONAL FUNCTION RECONSTRUCTION PROBLEM

ZACH OLESH and ARNE STORJOHANN

*David R. Cheriton School of Computer Science
University of Waterloo, Ontario, Canada N2L 3G1*

E-mail: astorjoh@uwaterloo.ca

<http://www.cs.uwaterloo.ca/~astorjoh/>

The final step of some algebraic algorithms is to reconstruct the common denominator d of a collection of rational functions v_*/d from their polynomial images modulo m . Using elementwise rational reconstruction requires that $\deg m > N + D$, where N and D are such that $\deg v_* \leq N$ and $\deg d \leq D$. We present an algorithm, based on minimal approximant basis computation, that can perform the reconstruction for many problem instances even when the modulus has considerably smaller degree, for example $\deg m > N + D/k$ for k a small constant.

Keywords: Rational function reconstruction

1. Introduction

Many algorithms in computer algebra that compute with polynomials from $\mathbb{K}[x]$, \mathbb{K} a field, use a homomorphic imaging scheme to avoid intermediate expression swell, to allow for simple course-grained parallelization, or to incorporate an output sensitive approach. Often, the last step of these algorithms is to reconstruct the common denominator d of a collection of rational functions $(v_i/d)_{1 \leq i \leq n}$ from their polynomial images $(u_i)_{1 \leq i \leq n}$ modulo m . The images modulo m are typically computed by combining multiple smaller images using either Chinese remaindering ($m = p_1 p_2 \cdots p_l$) or p -adic lifting ($m = p^l$).

Typically, the overall cost of an algorithm that uses homomorphic imaging depends on l , the number of images computed, which is directly related to $\deg m$. Ideally, the algorithm computes just enough images to allow reconstruction of the common denominator d . We first recall how elementwise rational function reconstruction can be applied, and then discuss our vector based variant that for some applications can save close to half of the

required image computations.

The rational function reconstruction problem takes as input a nonzero modulus $m \in \mathbb{K}[x]$, a single image polynomial $u \in \mathbb{K}[x]$ with $\deg u < \deg m$, and degree bounds $0 \leq N < \deg m$ and $0 \leq D < \deg m$. A solution to the problem is a pair of polynomials (d, v) such that

$$du \equiv v \pmod{m}, \quad \deg d \leq D, \quad \deg u \leq N. \quad (1)$$

If (d, v) is a solution to (1) that satisfies $\gcd(d, m) = 1$, then

$$u \equiv \frac{v}{d} \pmod{m}, \quad \deg d \leq D, \quad \deg u \leq N. \quad (2)$$

For convenience, in order to avoid some special cases, we have used the weaker condition (1) to define a solution to the problem rather than (2).

The vector generalization of the problem is defined similarly except with u replaced by $[u_1, \dots, u_n] \in \mathbb{K}[x]^{1 \times n}$. A solution to the vector version is then a pair $(d, [v_1, \dots, v_n])$ such that

$$d[u_1, u_2, \dots, u_n] \equiv [v_1, v_2, \dots, v_n] \pmod{m}, \quad \deg d \leq D, \quad \deg v_* \leq N. \quad (3)$$

Similarly, if $\gcd(d, m) = 1$ we have

$$[u_1, u_2, \dots, u_n] \equiv \left[\frac{v_1}{d}, \frac{v_2}{d}, \dots, \frac{v_n}{d} \right] \pmod{m}, \quad \deg d \leq D, \quad \deg v_* \leq N. \quad (4)$$

The link between solutions of (1) and certain rows of the traditional extended Euclidean algorithm has been well studied.¹ In general, we require $\deg m > N + D$ to ensure that the solution space is uniquely generated, that is, that every solution can be expressed as a polynomial multiple of a single generating solution (d, v) . Let $\text{Ratrecon}(u, m, N, D)$ denote a function that takes as input an instance of the problem with $\deg m > N + D$, and returns as output the first component d (possibly the zero polynomial) of a generating solution. The approach taken in various software libraries²⁻⁵ to compute the common d of the vector version of the problem is as follows:

Choose $N \geq 0$ and $D \geq 0$ such that $\deg m > N + D$;

$d := 1$;

for i **from** 1 **to** n **do**

$d := d \times \text{Ratrecon}(du_i \pmod{m}, m, N, D)$

od;

return d

The choice of N and D will depend on the particular application. Suppose that the v_i and d shown in (4) are the actual target solution to a

particular problem. On the one hand, if N and D are *a priori* bounds satisfying (4), then we know *a priori* that the output of the code fragment will be the same denominator d (up to normalization). On the other hand, if an output sensitive approach is being used, and N and D are guesses which may or may not satisfy (4), then the output must be assayed for correctness. If determined to be incorrect, the modulus m is augmented and the reconstruction attempted again. Implementations of Ratrecon, using either the algorithm of this paper or an approach based on half-gcd,¹ have running time bounded by $O(\mathbf{B}(\deg m))$ operations in \mathbf{K} , where \mathbf{B} is a cost function for gcd-like operations^a. Thus, the code fragment above will solve the vector version of the problem with $O(n \mathbf{B}(\deg m))$ operations in \mathbf{K} . Note that the running time for the reconstruction is pseudo-linear in the size of the input; in typical applications the cost of computing the images $[u_1, \dots, u_m] \bmod m$ will dominate, even to the extent that the time for the reconstruction is negligible in comparison. To save on the number of images that are computed and thus speed up the overall computation we must relax the condition $\deg m > N + D$.

Suppose $\deg m > N + D/k$ for some $k \in \mathbb{Z}_{>0}$. We present an algorithm that computes a complete basis of solutions to (3) using

$$O(nk^{\omega-1} \mathbf{B}(\deg m)) \quad (5)$$

operations in \mathbf{K} , where $2 \leq \omega \leq 3$ is a feasible exponent for matrix multiplication. By a basis we mean a set of solutions $(d^{(i)}, v^{(i)})_{1 \leq i \leq s}$, each $d^{(i)} \in \mathbf{K}[x]$ and $v^{(i)} \in \mathbf{K}[x]^{1 \times n}$, such that every solution admits a unique decomposition as a $\mathbf{K}[x]$ -linear combination of basis elements. The algorithm is similar to the approach based on Ratrecon above, except with the loop iterating only n/k times, each iteration dealing with a block of k images simultaneously. The approach works because we can show that the solution basis for all subproblems will have dimension bounded by k . Actually, for many problem instances the solution space will be uniquely generated ($s \leq 1$) whenever $\deg m > N + D/n$. Next we give an example of an application that generates such problem instances.

Suppose we want to compute $A^{-1}b \in \mathbf{K}(x)^{n \times 1}$ for a nonsingular $A \in \mathbf{K}[x]^{n \times n}$ and $b \in \mathbf{K}[x]^{n \times 1}$ from the image $A^{-1}b \bmod m$ for some m . For simplicity, assume that $\deg b = \deg A$. Let N be a bound for the degree of the numerators of $A^{-1}b$. For example, the *a priori* bound $N = n \deg A$ will be tight for a generic problem instance. From the assumption that

^aWe can take $\mathbf{B}(t) = M(t) \log t$ where M is a multiplication time for $\mathbf{K}[x]$, see [1, Def. 8.26].

$\deg b = \deg A$ it follows that the denominator of $A^{-1}b$ also has degree bounded by N . To apply elementwise reconstruction we need the image $A^{-1}b \bmod m$ with $\deg m > 2N$. However, in Ref. 6 it was observed that output of the vector rational reconstruction problem with input $A^{-1}b \bmod m$ will be uniquely generated whenever $\deg m > N + \deg A$. Thus, using the vector reconstruction algorithm it will suffice to have $\deg m > N + N/k$ for any $k \in \mathbb{Z}_{>0}$ that satisfies $N/k \geq \deg A$. For k a small constant, say $k = 5$, the reconstruction will still be relatively fast (compare with (5)) but the required lower bound $N + N/5$ for the modulus degree is a factor of 0.6 smaller than the bound $2N$ required for the elementwise approach.

We defined the rational function and vector rational function reconstruction problem to take as input bounds N and D . We remark that algorithms for a variant of the first problem called maximal quotient rational function reconstruction are given in Refs. 7,8. The maximal quotient problem takes as input u and m but not N and D , and returns as output the most likely candidate for v/d . The maximal quotient algorithms are useful in conjunction with an output sensitive approach when the difference between $\deg v$ and $\deg d$ may be large, but unknown. In particular, the approach is likely to succeed when $\deg m$ is modestly larger than $\deg v + \deg d$, compared to the required $\deg m > 2 \max(\deg v, \deg d)$ when a common bound $N = D$ is specified.

The rest of this paper is organised as follows. Sections 2 and 3 recall the notion of a reduced basis and minimal approximant bases. Section 3 also gives an algorithm for a special type of simultaneous matrix Padé approximation, the basis of the vector rational function reconstruction algorithm presented in Sec. 4. In Sec. 5 we show how the vector reconstruction algorithm may be applied to rational system solving over $\mathbb{K}[x]$. For more background on the definitions and concepts introduced in Secs. 2 and 3 we refer to Refs. 9–12. Fundamental notions and algorithms for polynomial matrices can be found in Refs. 13,14.

2. Reduced bases

Let $A \in \mathbb{K}[x]^{n \times m}$ have rank r . Let $\mathcal{L}(A)$ denote the lattice generated by the set of all $\mathbb{K}[x]$ -linear combinations of rows of A . In many applications we are interested in the subset of a lattice comprised of all rows $w \in \mathbb{K}[x]^{1 \times m}$ that satisfy a degree constraint specified by a fixed multi-index $\vec{n} = (n_1, n_2, \dots, n_m) \in \mathbb{Z}^m$:

$$w = [\overset{\leq n_1}{w_1}, \overset{\leq n_2}{w_2}, \dots, \overset{\leq n_m}{w_m}] \in \mathbb{K}[x]^{1 \times m} \quad (6)$$

Following [9, Def. 3.1], the *defect* of a row $w = [w_1, w_2, \dots, w_m] \in \mathbb{K}[x]^{1 \times m}$ with respect to \vec{n} is defined by

$$\text{dct}(w) = \text{dct}(w, \vec{n}) := \min_i \{n_i + 1 - \deg w_i\}, \quad (7)$$

where the zero polynomial has degree $-\infty$. The notion of defect measures the gap between w and the degree constraint \vec{n} : w satisfies (6) if and only if $\text{dct}(w)$ is positive. The following definition is similar to [11, Def. 5.1].

Definition 2.1. A matrix $B = [b_1^T | b_2^T | \dots | b_r^T]^T \in \mathbb{K}[x]^{r \times m}$ is a *reduced basis* of type \vec{n} for $A \in \mathbb{K}[x]^{n \times m}$ if the following conditions are satisfied:

- (i) B has full row rank and $\mathcal{L}(B) = \mathcal{L}(A)$. [**basis property**]
- (ii) Each $w \in \mathcal{L}(B)$ admits a unique decomposition $w = \sum_{i=1}^r c_i b_i$ with $c_i \in \mathbb{K}[x]$, $\deg c_i \leq \text{dct}(b_i) - \text{dct}(w)$, $1 \leq i \leq r$. [**reduced property**]

The reduced bases are precisely those with maximal defect.

By *positive part* of a reduced basis we mean the submatrix comprised of the rows with positive defect. All $w \in \mathcal{L}(A)$ that satisfy the degree constraint \vec{n} are generated by the positive part of a reduced basis for A : if $\text{dct}(b_i) \leq 0$ and $\text{dct}(w) > 0$, then the c_i of Def. 2.1 has $\deg c_i \leq \text{dct}(b_i) - \text{dct}(w) < 0$ and thus c_i is the zero polynomial.

Suppose B is a basis for A , rows permuted so that defects are nonincreasing. Then reduced bases are precisely those with $(\text{dct}(b_1), \dots, \text{dct}(b_r))$ lexicographically maximal among all bases for A whose rows are similarly permuted. Thus, up to row permutation, any two reduced bases of type \vec{n} for A will have the same tuple of defects. It follows that the number of rows in the positive part of a reduced basis is an invariant of A .

3. Minimal approximant bases

Let $G \in \mathbb{K}[x]^{n \times m}$, $\vec{n} \in \mathbb{Z}^n$, and $d \in \mathbb{Z}_{\geq 0}$.

Definition 3.1. An order d *minimal approximant* of type \vec{n} for G is a reduced basis M of type \vec{n} for the lattice $\{w \in \mathbb{K}[x]^{1 \times n} \mid wG \equiv 0 \pmod{x^d}\}$.

Note that a minimal approximant M as in Def. 3.1 will necessarily have dimension $n \times n$, be nonsingular, and satisfy $MG \equiv 0 \pmod{x^d}$.

The following is restatement of [15, Theorem 2.4]. We remark that Ref. 15 gives more precise cost estimates in terms of certain ad hoc cost functions. We will use the exponent ω and cost function B .

Theorem 3.1. *There exists an algorithm `MinBasis` that takes as input $(G, d, \vec{n}) \in (\mathbb{K}[x]^{n \times m}, \mathbb{Z}_{\geq 0}, \mathbb{Z}^n)$ and returns as output $(M, \delta) \in$*

$(\mathbb{K}[x]^{n \times n}, \mathbb{Z}^n)$, an order d minimal approximant M of type \vec{n} for G together with a tuple $\delta = (\delta_1, \dots, \delta_n)$ of the defects of rows of M . If $m \leq n$, the cost of the algorithm is $O(n^\omega \mathbf{B}(d))$ operations in \mathbb{K} .

For brevity, we will say that (M, δ) in Theorem 3.1 solves the minimal approximant problem with input (G, d, \vec{n}) . By $\text{PosMinBasis}(G, d, \vec{n})$ we mean the output of $\text{MinBasis}(G, d, \vec{n})$ restricted to the rows with positive defect; this may be a $0 \times n$ matrix.

We now give two technical lemmas that follow from the definition of minimal approximant and the properties of reduced bases. The first lemma states that zero rows in an input matrix can be ignored as far as minimal approximant basis computation is concerned.

$$\left[\begin{array}{c|c} M & H \\ \hline & I_k \end{array} \right] \begin{bmatrix} * \\ - \end{bmatrix} \equiv 0 \pmod{x^d}$$

Lemma 3.1. *Let $H \in \mathbb{K}[x]^{n \times m}$ have its last k rows zero and let $\vec{n} = (n_1, \dots, n_n)$. If $M \in \mathbb{K}[x]^{(n-k) \times (n-k)}$ is an order d minimal approximant of type (n_1, \dots, n_{n-k}) for the first $n - k$ row of H , then $\text{diag}(M, I_k)$ is an order d the minimal approximant of type \vec{n} for H .*

The next lemma follows as a special case of [10, Theorem 5.1], which gives a general result regarding the recursive computation of minimal approximants. Let $\mathbf{1}$ denote the tuple $(1, 1, \dots, 1)$ of appropriate length.

Lemma 3.2. *Let $H \in \mathbb{K}[x]^{n \times m}$ and $H' \in \mathbb{K}[x]^{n \times m'}$. If $(M, \delta) := \text{MinBasis}(H, d, \vec{n})$ and $(M', \delta') := \text{MinBasis}(MH', d, \delta - \mathbf{1})$, then $(M'M, \delta')$ solves the minimal approximant problem with input $([H|H'], d, \vec{n})$.*

The $-\mathbf{1}$ in the second call to MinBasis in Lemma 3.2 is due to the $+1$ in the definition of defect (see (7)). For example, in the special case where H is the zero matrix, an order d minimal approximant of type \vec{n} for H is given by I_n , with row defects $\delta = \vec{n} + \mathbf{1}$. For more details we refer to [10, Sections 3 and 4].

As noted after Def. 2.1, if $w \in \mathcal{L}(\text{MinBasis}(H, d, \vec{n}))$ has positive defect with respect to \vec{n} , then $w \in \mathcal{L}(\text{PosMinBasis}(H, d, \vec{n}))$. Since $\mathcal{L}(M'M) \subseteq \mathcal{L}(M)$, any row in $M'M$ with positive defect with respect to \vec{n} is comprised of a linear combination of rows of $\text{PosMinBasis}(H, d, \vec{n})$. We get the following as a corollary.

Corollary 3.1. *Lemma 3.2 still holds if MinBasis is replaced by PosMinBasis and “minimal approximant” is replaced by “positive part min-*

imal approximant.”

3.1. An algorithm for simultaneous Padé approximation

We describe an algorithm to compute an order d minimal approximant of type \vec{n} for an input matrix G that can be decomposed as

$$G = \left[\begin{array}{c|c|c|c|c} G_1 & G_2 & \cdots & G_n & \\ \hline E & & & & \\ \hline & E & & & \\ \hline & & \ddots & & \\ \hline & & & & E \end{array} \right] \in \mathbb{K}[x]^{(m+tn) \times nk}, \quad (8)$$

each $G_i \in \mathbb{K}[x]^{m \times k}$ and $E \in \mathbb{K}[x]^{t \times k}$. We will assume that $\vec{n} = (\vec{n}_1, \vec{n}_2, \dots, \vec{n}_n)$ with $\vec{n}_1 \in \mathbb{Z}_{\geq 0}^m$ and $\vec{n}_2 \in \mathbb{Z}_{\geq 0}^t$, but remark that the algorithm we present can be adapted to work for an arbitrary degree constraint $\vec{n} \in \mathbb{Z}^{m+tn}$. Actually, our goal is to compute only the first m columns of the positive part of an order d minimal approximant of type \vec{n} . Lemma 3.1 and Corollary 3.1 suggest an iterative approach that works in stages for $i = 1, 2, \dots, n$. The approach can be understood by considering stage 2. Suppose we have the first m columns $\bar{M} \in \mathbb{K}[x]^{s \times m}$ of the positive part $[\bar{M} | *] \in \mathbb{K}[x]^{s \times (m+t)}$ of an order d minimal approximant of type (\vec{n}_1, \vec{n}_2) for

$$\left[\begin{array}{c} G_1 \\ E \end{array} \right] \in \mathbb{K}[x]^{(m+t) \times k},$$

together with a corresponding tuple $\delta \in \mathbb{Z}_{>0}^s$ of defects. By Lemma 3.1, $\text{diag}([\bar{M} | *], I_t)$, with defect tuple $(\delta, \vec{n}_2 + \mathbf{1})$, is the the positive part of an order d minimal approximant of type $(\vec{n}_1, \vec{n}_2, \vec{n}_2)$ for the first k columns H of

$$[H | H'] = \left[\begin{array}{c|c} G_1 & G_2 \\ \hline E & \\ \hline & E \end{array} \right] \in \mathbb{K}[x]^{(m+2t) \times 2k}. \quad (9)$$

By Corollary 3.1, if

$$(M', \delta') := \text{PosMinBasis}(\text{diag}([\bar{M} | *], I_t)H', d, (\delta, \vec{n}_2 + \mathbf{1}) - \mathbf{1}),$$

then $M' \text{diag}([\bar{M} | *], I_t)$ will be the positive part of an order d minimal approximant of type $(\vec{n}_1, \vec{n}_2, \vec{n}_2)$ for $[H | H']$. The key observation is that the first argument of `PosMinBasis` is given by

$$\left[\begin{array}{c|c} \bar{M} * & \\ \hline & I_t \end{array} \right] \left[\begin{array}{c} G_2 \\ E \end{array} \right] = \left[\begin{array}{c} \bar{M}G_2 \\ E \end{array} \right],$$

so we don't need to know the unknown block $*$ of $[\bar{M} \mid *]$. Once M' is computed, the first m columns of the positive part of a minimal approximant for $[H \mid H']$ can be computed as $M'\bar{M}$. Stages $i = 3, 4, \dots, n$ are similar. This gives the following algorithm.

Algorithm: `SimPade`($[G_1, \dots, G_n], E, d, \vec{n}_1, \vec{n}_2$)

Input: $G_* \in \mathbb{K}[x]^{m \times k}$, $E \in \mathbb{K}[x]^{t \times k}$, $d \in \mathbb{Z}_{\geq 0}$, $\vec{n}_1 \in \mathbb{Z}_{\geq 0}^m$, $\vec{n}_2 \in \mathbb{Z}_{\geq 0}^t$.

Output: (\bar{M}, δ) , \bar{M} the first m columns of an M such that (M, δ) is a valid output of `PosMinBasis`($G, d, (\vec{n}_1, \vec{n}_2, \dots, \vec{n}_2)$), with G as in (8).

$(\bar{M}, \delta) := (I_m, \vec{n}_1 + \mathbf{1})$;

for i **from** 1 **to** n **do**

$\delta := (\delta, \vec{n}_2 + \mathbf{1})$;

$(M', \mu) := \text{PosMinBasis} \left(\begin{bmatrix} \bar{M}G_i \\ E \end{bmatrix}, d, \delta - \mathbf{1} \right)$;

$\bar{M} := M'\bar{M}$

od;

return (\bar{M}, δ)

The cost of algorithm `SimPade` will depend on the row dimensions of the first argument to the n calls to `PosMinBasis`. In the next section we will see that for some inputs to the algorithm we can be sure that \bar{M} will never have more than k rows.

Theorem 3.2. *Algorithm `SimPade` is correct. If $t = O(k)$ and the dimension of \bar{M} remains bounded by k throughout, the cost of the algorithm is $O((nk + m)k^{\omega-1} \mathbf{B}(d))$ operations in \mathbb{K} .*

4. Vector rational function reconstruction

Fix the following quantities throughout this section:

- a nonzero modulus $m \in \mathbb{K}[x]$,
- an input vector $u \in \mathbb{K}[x]^{1 \times n}$ with $\deg u < \deg m$, and
- degree bounds N and D with $0 \leq N < \deg m$ and $0 \leq D < \deg m$.

A vector $[d \mid v] \in \mathbb{K}[x]^{1 \times (n+1)}$ ($d \in \mathbb{K}[x]$, $v \in \mathbb{K}[x]^{1 \times n}$) solves the vector rational function reconstruction problem if $du \equiv v \pmod{m}$, with $\deg d \leq D$ and $\deg v \leq N$. The complete set of solutions is thus

$$\mathcal{S} = \{[d \mid v] \in \mathbb{K}[x]^{1 \times (n+1)} \mid du \equiv v \pmod{m}, \deg d \leq D, \deg v \leq N\}.$$

Consider the lattice generated by the nonsingular matrix

$$A = \left[\begin{array}{c|c} 1 & u \\ \hline & mI_n \end{array} \right] \in \mathbb{K}[x]^{(n+1) \times (n+1)}. \quad (10)$$

Any vector in $\mathcal{L}(A)$ with degree strictly less than $\deg m$ has the form $[d | du \bmod m]$ for $d \in \mathbb{K}[x]$ with $\deg d < \deg m$: the rows of A containing mI_n serve to reduce modulo m the last n entries in $d [1 | u]$. If we set degree constraints (D, N, \dots, N) , then $[d | v] \in \mathcal{S}$ if and only if $[d | v] \in \mathcal{L}(A)$ with $\text{dct}([d | v]) > 0$. Thus, \mathcal{S} is generated by the positive part $B = [b_1^T | b_2^T | \dots | b_s^T]^T \in \mathbb{K}[x]^{s \times (n+1)}$ of a reduced basis of type (D, N, \dots, N) for A .

Theorem 4.1. $\mathcal{S} = \{\sum_{i=1}^s c_i b_i \mid c_i \in \mathbb{K}[x], \deg c_i < \text{dct}(b_i), 1 \leq i \leq s\}$.

Corollary 4.1. *If $e \in \mathbb{K}[x]^{s \times 1}$ is the first column of the positive part of a reduced basis of type (D, N, \dots, N) for A , then $[e | eu \bmod m] \in \mathbb{K}[x]^{s \times n}$ is the positive part of a reduced basis of type (D, N, \dots, N) for A .*

The next theorem gives an *a priori* upper bound on s , the number of rows in the positive part of a reduced basis of type (D, N, \dots, N) for A . Since the bound does not depend on n , it also applies for the number of rows in the positive part of a reduced basis of type (D, N, \dots, N) for the leading $j \times j$ submatrix of A , for any j with $2 \leq j \leq n + 1$.

Theorem 4.2. $s \leq k$ for $k \in \mathbb{Z}_{>0}$ minimal such that $\deg m > N + D/k$.

Proof. Assume for now that $N \geq D$. Then R is a reduced basis of type (D, N, \dots, N) for A if and only if $R' := R \text{diag}(x^{N-D}, I_n)$ is a reduced basis of type (N, N, \dots, N) for $A' := A \text{diag}(x^{N-D}, I_n)$. Thus, s is equal to the number of rows in R' with degree at most N . A reduced basis of type (N, N, \dots, N) for A' will have degree at most $\deg A'$, so $\deg R' \leq \deg A' = \deg m$. Using the fact that the determinant of a polynomial matrix is bounded by the sum of the row degrees now gives

$$\deg \det R' \leq sN + (n + 1 - s) \deg m. \quad (11)$$

Using the fact that $\det R'$ is a scalar multiple of $\det A'$ gives

$$\deg \det R' = \deg \det A' = N - D + n \deg m. \quad (12)$$

Combining (11) and (12) and solving for $\deg m$ gives

$$\deg m \leq N + \frac{D}{s-1}.$$

It follows that $s - 1 < k$. The case $D > N$ is similar. □

Let

$$G = \begin{bmatrix} u \\ -I_n \\ mI_n \end{bmatrix} \in \mathbb{K}[x]^{(2n+1) \times n}.$$

Dependant on the assumption that $\deg u < \deg m$, each $[d | v] \in \mathcal{S}$ can be extended with $r := -(du - v)/m \in \mathbb{K}[x]^{1 \times n}$ such that $\deg r \leq D - 1$ and $[d | v | r]G = 0$. Conversely, if $[d | v | r] \in \mathbb{K}[x]^{1 \times (2n+1)}$ satisfies $[d | v | r]G \equiv 0 \pmod{x^{D+\deg m}}$ and $(\deg d, \deg v, \deg r) \leq (D, N, D-1)$, then $du - v + mr \equiv 0 \pmod{x^{D+\deg m}}$ with $\deg(du - v + mr) < D + \deg m$, implying $du - v + mr = 0$ and thus $[d | v] \in \mathcal{S}$. Thus, the first $n + 1$ columns of the positive part of an order $D + \deg m$ minimal approximant of type

$$(D, N, \dots, N, D - 1, \dots, D - 1) \tag{13}$$

for G is a reduced basis of type (D, N, \dots, N) for A . By Corollary 4.1, it will suffice to compute only the first column of such a minimal approximant.

To apply algorithm **SimPade** we need to adjust the matrix G slightly. Let k be either n or as in Theorem 4.2, whichever is minimal. Assume for now that k divides n and write $u = [u_1 | u_2 | \dots | u_{n/k}]$, each $u_* \in \mathbb{K}[x]^{1 \times k}$. Permute the last $2n$ rows of G so that

$$G = \begin{bmatrix} u_1 & u_2 & \cdots & u_{n/k} \\ -I_k & & & \\ mI_k & & & \\ & -I_k & & \\ & mI_k & & \\ & & \ddots & \\ & & & -I_k \\ & & & mI_k \end{bmatrix} \in \mathbb{K}[x]^{(2n+1) \times n}.$$

In the special case when m is a power of x , the vector rational reconstruction problem is a simultaneous Padé approximation problem: the positive part of a reduced basis for A shown in (10) is the positive part of an order $\deg m$ minimal approximant of type (D, N, \dots, N) for

$$G = \begin{bmatrix} u_1 & u_2 & \cdots & u_{n/k} \\ -I_k & & & \\ & -I_k & & \\ & & \ddots & \\ & & & -I_k \end{bmatrix} \in \mathbb{K}[x]^{(n+1) \times n}.$$

This shows correctness of the following algorithm.

Algorithm: `VectorRecon`(u, m, N, D)

Input: $u \in \mathbb{K}[x]^{1 \times n}$, nonzero $m \in \mathbb{K}[x]$, $N \in \mathbb{Z}_{\geq 0}$, $D \in \mathbb{Z}_{\geq 0}$.

Output: An $e \in \mathbb{K}[x]^{s \times 1}$ as in Corollary 4.1.

Condition: $N < \deg m$, $D < \deg m$, $\deg u < \deg m$.

$k := \min\{n, \min\{t \in \mathbb{Z}_{>0} \mid \deg m > N + D/t\}\}$;

Augment u with at most $k - 1$ zeros so that $k \mid n$;

if $m = x^{\deg m}$ **then**

$E := -I_k$;

$\vec{n}_2 := (N, \dots, N)$;

$d := \deg m$

else

$E := \begin{bmatrix} -I_k \\ mI_k \end{bmatrix}$;

$\vec{n}_2 := (N, \dots, N, D - 1, \dots, D - 1)$;

$d := D + \deg m$

fi;

$\vec{n}_1 := (D)$;

Write $u = [u_1 \mid u_2 \mid \dots \mid u_{n/k}]$, each $u_i \in \mathbb{K}[x]^{1 \times k}$;

$(e, *) := \text{SimPade}([u_1, u_2, \dots, u_n], E, d, \vec{n}_1, \vec{n}_2)$;

Normalize each entry in e to be monic;

return e

Theorem 4.3. *Algorithm `VectorRecon` is correct. The cost of the algorithm is $O(nk^{\omega-1} \mathbf{B}(\deg m))$ operations in \mathbb{K} , where $k \in \mathbb{Z}_{>0}$ is minimal such that $\deg m > N + D/k$.*

5. Application to linear solving

Let a nonsingular $A \in \mathbb{K}[x]^{n \times n}$ and $b \in \mathbb{K}[x]^{n \times 1}$ be given. Let $d \in \mathbb{K}[x]$ be the denominator of $A^{-1}b$, that is, the minimal degree monic polynomial such that $v := dA^{-1}b$ is over $\mathbb{K}[x]$. One of the most effective methods to compute d is to iteratively compute

$$u := A^{-1}b \bmod p^l = c_0 + c_1p + c_2p^2 + \dots + c_{l-1}p^{l-1}, \quad (14)$$

each $c_i \in \mathbb{K}[x]^{n \times 1}$ with $\deg c_i < \deg p$, for larger and larger l using p -adic lifting^{16,17} for some p with $\gcd(p, \det A) = 1$, and then apply rational reconstruction. If desired, v can be computed as $du \bmod m$ once d is found. In the following theorem m plays the role of p^l .

Theorem 5.1. *If $\deg m > \max(N + \deg A, D + \deg b)$ and $u = A^{-1}b \bmod m$ then the output of `VectorRecon`(u^T, m, N, D) is either:*

- (i) $e = [d] \in \mathbb{K}[x]^{1 \times 1}$, if $N \geq \deg v$ and $D \geq \deg d$, or
- (ii) $e \in \mathbb{K}[x]^{0 \times 1}$, if at least one of $N < \deg v$ or $D < \deg d$.

Proof. Suppose $e = [e_1, e_2, \dots, e_s]^T \in \mathbb{K}[x]^{s \times 1}$ is the output of `VectorRecon`, and for $1 \leq i \leq s$ let $v_i := e_i A^{-1}b \bmod m$. The s vectors $[e_i \mid v_i^T] \in \mathbb{K}[x]^{1 \times (n+1)}$ are linearly independent and satisfy $Av_i \equiv e_i b \bmod m$. Since $\max(\deg Av_i, \deg e_i b) \leq \max(N + \deg A, D + \deg b) < \deg m$, we actually have $Av_i = e_i b$. Parts (i) and (ii) now follow by noting that the dimension of the solution space for these cases are 1 and 0, respectively. \square

Suppose N and D are *a priori* bounds: $N \geq \deg v$ and $D \geq \deg d$. Standard rational function reconstruction¹ can be used to recover d in $O(n \mathbf{B}(\deg m))$ field operations but requires $\deg m > N + D$. By Theorem 5.1, Algorithm `VectorRecon` can recover d in $O(nk^{\omega-1} \mathbf{B}(\deg m))$ field operations where $\deg m > \max(N + \deg A, D + \deg b, N + D/k)$.

Algorithm `VectorRecon` can also be used in conjunction with an output sensitive approach. Let $m = p^l$ and suppose we have u as in (14). Set \bar{N} to be the maximal integer such that $\deg m > \bar{N} + \max(\deg A, \deg b, \bar{N}/k)$. According to Theorem 5.1, the call `VectorRecon`(u^T, m, \bar{N}, \bar{N}) will either recover the denominator d or determine that $\max(\deg d, \deg v) > \bar{N}$.

6. Conclusion

The approach we have described here for reconstructing a vector of rational functions with common denominator can be adapted to the problem of reconstructing a vector of rational numbers with a common denominator. This requires the use of integer lattice basis reduction¹⁸ and will be described in a future paper.

References

1. J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, 2 edn. (Cambridge University Press, 2003).
2. Z. Chen and A. Storjohann, A BLAS based C library for exact linear algebra on integer matrices, in *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '05*, ed. M. Kauers (ACM Press, New York, 2005).
3. J.-G. Dumas, T. Gautier, M. Giesbrecht, P. Giorgi, B. Hovinen, E. Kaltofen, B. D. Saunders, W. J. Turner and G. Villard, LinBox: A generic library for exact linear algebra., in *Proc. First Internat. Congress Math. Software ICMS*

- 2002, Beijing, China, eds. A. J. Cohen and N. Gao, X.-S. and Takayama (World Scientific, Singapore, 2002).
4. P. Giorgi, Arithmetic and algorithmic in exact linear algebra for the LinBox library, PhD thesis, Ecole normale supérieure de Lyon, LIP, (Lyon, France, 2004).
 5. V. Shoup, *NTL: A Library for Doing Number Theory*, (2005). <http://www.shoup.net/ntl/>.
 6. T. Mulders and A. Storjohann, Rational solutions of singular linear systems, in *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '00*, ed. C. Traverso (ACM Press, New York, 2000).
 7. S. Khodadad and M. Monagan, Fast rational function reconstruction, in *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '05*, ed. J.-G. Dumas (ACM Press, New York, 2006).
 8. M. Monagan, Maximal quotient rational reconstruction: an almost optimal algorithm for rational reconstruction, in *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '04*, ed. J. Gutierrez (ACM Press, New York, 2004).
 9. B. Beckermann and G. Labahn, *SIAM Journal on Matrix Analysis and Applications* **15**, 804 (1994).
 10. B. Beckermann and G. Labahn, *Journal of Computational and Applied Math* **77**, 5 (1997).
 11. B. Beckermann, G. Labahn and G. Villard, Shifted normal forms of polynomial matrices, in *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '99*, ed. S. Dooley (ACM Press, New York, 1999).
 12. B. Beckermann, G. Labahn and G. Villard, *Normal Forms for General Polynomial Matrices*, Research Report 2002-1, ENS Lyon (France, 2002).
 13. D. Bini and V. Y. Pan, *Polynomial and Matrix Computations, Vol 1: Fundamental Algorithms* (Birkhauser, Boston, 1994).
 14. T. Kailath, *Linear Systems* (Prentice Hall, Englewood Cliffs, N.J., 1980).
 15. P. Giorgi, C.-P. Jeannerod and G. Villard, On the complexity of polynomial matrix computations, in *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '03*, ed. R. Sendra (ACM Press, New York, 2003).
 16. J. D. Dixon, *Numer. Math.* **40**, 137 (1982).
 17. R. T. Moenck and J. H. Carter, Approximate algorithms to derive exact solutions to systems of linear equations., in *Proc. EUROSAM '79, volume 72 of Lecture Notes in Compute Science*, (Springer-Verlag, Berlin-Heidelberg-New York, 1979).
 18. A. K. Lenstra, H. W. Lenstra and L. Lovász, *Math. Ann.* **261**, 515 (1982).