Algorithms for Simultaneous Hermite Padé Approximations*

Johan Rosenkilde, né Nielsen
Technical University of Denmark
Denmark
jsrn@jsrn.dk

Arne Storjohann
University of Waterloo
Canada
astorjoh@uwaterloo.ca

Abstract

We describe how to solve simultaneous Hermite Padé approximations, sometimes known simply as Hermite Padé, over a polynomial ring $\mathsf{K}[x]$ for a field K using $O^{\sim}(n^{\omega-1}td)$ operations in K, where d is the sought precision, and where n is the number of simultaneous approximations using $t < n$ polynomials. We develop two algorithms using different approaches. Both algorithms return a reduced sub-basis that generates the complete set of solutions to the input approximations problem that satisfy the given degree constraints. Previously, the cost $O^{\sim}(n^{\omega-1}td)$ has only been reached with algorithms finding a single solution for the case $t < n$. Our results are made possible by recent breakthroughs in fast computations of minimal approximant basis and Hermite Padé approximations for the case $t \geq n$.

1 Introduction

Let K be a field admitting exact computation. Padé approximation concerns approximating a power series $S \in \mathsf{K}[[x]]$ with a rational function $\frac{\phi}{\lambda}$ up to some prescribed precision d, while keeping the degrees of ϕ and λ small, i.e., such that $\lambda S \equiv \phi \mod x^d$. There are two natural vector-generalisations to this:

Simultaneous Padé approximation is where we have several power series $S_1, \ldots, S_n \in \mathsf{K}[[x]]$ and seek rational functions $\frac{\phi_1}{\lambda}, \ldots, \frac{\phi_n}{\lambda}$, all sharing the same denominator λ, and such that $\lambda S_i \equiv \phi_i \mod x^d$ for each i. In vector form:

$$\lambda \begin{bmatrix} S_1 & S_2 & \cdots & S_n \end{bmatrix} \equiv \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_n \end{bmatrix} \bmod x^d.$$

Hermite Padé approximation is where we have a several power series $S_1, \ldots, S_t \in \mathsf{K}[[x]]$ and seek several polynomials $\lambda_1, \ldots, \lambda_t$ and a single ϕ

such that $\lambda_1 S_1 + \ldots + \lambda_n S_n \equiv \phi \mod x^d$. In vector form:

$$\begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_t \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_t \end{bmatrix} \equiv \phi \bmod x^d.$$

The study of both generalisations trace back to Hermite's proof of the transcendence of e [18], and were subsequently studied in greater detail by Padé [28]. There is a duality between the solution sets of the two problems, first observed by Mahler [25]. See also [1] for a detailed treatment of Padé approximations and these generalisations over the real or complex numbers.

From the study of these and other type of approximants emerged unifying generalisations, e.g. [2, 3, 5]. One form of these is what we will call simultaneous Hermite Padé approximations of size $t \times n$:

> Given a matrix $\boldsymbol{S} \in \mathsf{K}[x]^{t \times n}$ find two low-degree vectors $\lambda \in \mathsf{K}[x]^{1 \times t}$ and $\boldsymbol{\phi} \in \mathsf{K}[x]^{1 \times n}$, such that $\boldsymbol{\lambda S} \in \boldsymbol{\phi} \bmod x^d$. In matrix form:
>
> $$\begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_t \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1n} \\ S_{21} & S_{22} & \cdots & S_{2n} \\ \vdots & \vdots & & \vdots \\ S_{t1} & S_{t2} & \cdots & S_{tn} \end{bmatrix} \equiv \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_n \end{bmatrix} \bmod x^d$$

Note that the boundary cases $t = 1$ (with n arbitrary) and $n = 1$ (with t arbitrary) are the simultaneous Padé and Hermite Padé approximation problems, respectively. We also remark that there is a continuum of problems depending on the relation between t and n. For $t < n$ the matrix $\boldsymbol{S}$ is a "fat" row vector, suggesting a problem closer to simultaneous Padé, while for $t > n$ the matrix $\boldsymbol{S}$ is a "fat" column vector, closer to Hermite Padé.

Our focus is where K admits exact computation, especially K being a finite field, and for the case $t < n$. While the new algorithms we propose in this paper are applicable if $t \geq n$, they are designed to give improved complexity estimates compared to previous approaches for the case $t < n$. Assuming $t < n$, we give new deterministic algorithms with cost $O^{\sim}(n^{\omega-1}td)$. This matches the previously best cost which uses a randomized algorithm, see below. Furthermore, our algorithms produce a parametrisation of *all* solutions, which the randomized algorithm does not seem amenable to.

Numerous earlier algorithms exist for the simultaneous Hermite Padé approximation problem as well as for very related problems which can be applied to this. Specializing cost estimates to the case $t < n$, Beckermann and Labahn [5] obtain both $O(n^2td^2)$ and $O^{\sim}(n^3d)$; see also there for a discussion of the other approaches of the early 90s; their approach is very similar to

the extended euclidean algorithm and its fast variant. Zeh et. al [34] obtain $O(nt^2d^2)$; this approach is inspired by the Berlekamp–Massey algorithm [9].

The problem can also be solved using fast minimal approximant basis, or order basis: for $t < n$ the cost is $O^{\sim}(n^{\omega}d)$ using e.g. [14, 15]; this approach traces back to [2, 4]. Note that most of these approaches can actually produce a parametrisation of *all* solutions. See also the related concepts interpolation basis [20] and relation basis [27].

A completely different approach is to write the approximation as a linear system and remark that it has small displacement rank $n + t$; in fact, classical Padé approximations and similar Toeplitz systems was one of the main inspirations for developing displacement rank algorithms. Using [10] this would have[1] the complexity $O^{\sim}(n^{\omega-1}td)$ for $t < n$, though randomized. Further, this algorithm can not in a straightforward way be used to produce all solutions.

Now consider the case $t \geq n$. Although the algorithms we give in this paper are designed to be fast for the case $t < n$, we exploit the natural duality between the simultaneous and Hermite Padé approximation problems and rely on algorithms that are fast in the case $t \geq n$. In particular, using the fast minimal approximant basis algorithms for input matrices with more rows than columns [22, 35], gives an algorithm for the simultaneous Hermite Padé approximation problem with cost only $O^{\sim}(t^{\omega-1}nd)$ if $t \geq n$.

Let us now give a formal description of the problem. In the previous discussion, we assumed for simplicity that the moduli where all equal to x^d. We actually consider a generalisation where the moduli are replaced by arbitrary polynomials g_i. The problem is formalised as Problem 1.1.

Problem 1.1 ($(t \times n)$ simultaneous Hermite Padé)**.**
Given a tuple $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$ *where*

- $\boldsymbol{S} = [\boldsymbol{S}_1 \mid \ldots \mid \boldsymbol{S}_n] \in \mathsf{K}[x]^{t \times n}$ *a matrix of polynomials with columns* $\boldsymbol{S}_i$*.*
- $\boldsymbol{g} = (g_1, \ldots, g_n) \in \mathsf{K}[x]^n$ *is a sequence of moduli polynomials with* $\deg \boldsymbol{S}_i < \deg g_i$ *for* $i = 1, \ldots, n$*,*
- *and* $\boldsymbol{N} = (T_1, \ldots, T_t, N_1, \ldots, N_n) \in \mathbb{Z}_{\geq 0}^{t+n}$ *are degree bounds satisfying* $1 \leq T_j \leq \deg \operatorname{lcm}(g_1, \ldots, g_n) + 1$ *for* $j = 1, \ldots, t$ *and* $N_i \leq \deg g_i$ *for* $i = 1, \ldots, n$*,*

find, if it exists, a non-zero vector $(\lambda_1, \ldots, \lambda_t, \phi_1, \ldots, \phi_n)$ *such that*

1. $(\lambda_1, \ldots, \lambda_t)\boldsymbol{S}_i \equiv \phi_i \mod g_i$ *for* $i = 1, \ldots, n$*, and*

[1] Before applying the algorithm of [10], one needs to compute "generators" of the displacement-representation of the system. We do not assert that this can be done sufficiently fast, and it is outside the scope of this related work section, but it is likely true. Note that in the earlier version of this paper [29], we erroneously claimed that the cost of applying [10] to the case $t = 1$ would cost $O^{\sim}(n^{\omega}d)$.

2. $\deg \lambda_j < T_j$ *for* $j = 1, \ldots, t$ *and* $\deg \phi_i < N_i$ *for* $i = 1, \ldots, n$.

We will call any vector $(\lambda_1, \ldots, \lambda_t, \phi_1, \ldots, \phi_n)$ as above *a solution* to a given simultaneous Hermite Padé approximation problem. Note that if the entries of $\boldsymbol{N}$ are set too low, then it might be the case that no solution exists.

Example 1.2. Consider over $\mathbb{F}_2[x]$ that $g_1 = g_2 = g_3 = x^5 - 1$, and

$$\boldsymbol{S} = \begin{bmatrix} x^4 + x^2 + 1 & x^4 + x & x^4 + x^2 & x^4 + x^2 + x + 1 \\ x^4 + x + 1 & x^4 + x^3 + 1 & x^4 + x^2 + x + 1 & x^4 + x^3 + x^2 + 1 \end{bmatrix} ,$$
$$\boldsymbol{N} = (T_1, T_2, N_1, N_2, N_3, N_4) = (5, 3, 2, 3, 4, 4) .$$

Then $\boldsymbol{\lambda}_1 = \left(x^4 + x^3 + x,\ x^2 + 1\right)$ is a solution, since $\deg \lambda_{11} < 5$, $\deg \lambda_{12} < 3$ and

$$\boldsymbol{\lambda}_1 \boldsymbol{S} \equiv \left(1,\ x^2 + x,\ x^3 + x^2 + x + 1,\ x + 1\right) \mod x^5 - 1 .$$

$\boldsymbol{\lambda}_2 = \left(x^3 + x,\ x\right)$ is another solution, since

$$\boldsymbol{\lambda}_2 \boldsymbol{S} \equiv \left(1,\ x,\ x + 1,\ x^3 + 1\right) \mod x^5 - 1 .$$

These two solutions are linearly independent over $\mathbb{F}_2[x]$ and span all solutions.

We remark on the upper bound $T_i \leq \deg \operatorname{lcm}(g_1, \ldots, g_n) + 1$. For some cases, this bound is attained: take e.g. $t = 1$, $S_1 = \ldots = S_n = 1$, and $N_i = 0$ for all i, and all g_i pairwise coprime. This stipulates that λ should be divisible by all g_i, i.e. $\lambda = g_1 g_2 \cdots g_n$ the smallest possible solution. Conversely, this bound is always sufficient: notice that taking $\boldsymbol{\lambda} = (\operatorname{lcm}(g_1, g_2, \ldots, g_n), 0, 0, \ldots, 0)$ yields $\boldsymbol{\lambda} \boldsymbol{S}_i \equiv 0 \mod g_i$ for $i = 1, \ldots, n$, satisfying any degree bounds $N_1, \ldots, N_n$.

As mentioned earlier, several previous algorithms for solving Problem 1.1 and some of its special cases are more ambitious and produce an entire *basis* of solutions that satisfy the first output condition $(\lambda_1, \ldots, \lambda_t)\boldsymbol{S}_i \equiv \phi_i \mod g_i$ for $i = 1, \ldots, n$, including solutions that do not satisfy the degree bounds stipulated by the second output condition. Our algorithms are slightly more restricted in that we only return the sub-basis that generates the set of solutions that satisfy both output requirements of Problem 1.1. Formally:

Problem 1.3 ($(t \times n)$ simultaneous Hermite Padé basis)**.**
Given an instance of Problem 1.1, find a matrix $A \in \mathsf{K}[x]^{* \times (t+n)}$ *such that:*

- *Each row of* A *is a solution to the instance.*
- *All solutions are in the* $\mathsf{K}[x]$*-row space of* A*.*
- A *is* $(-\boldsymbol{N})$*-row reduced*[2]*.*

[2]The notions $(-\boldsymbol{N})$-degree, $\deg_{(-\boldsymbol{N})}$ and $(-\boldsymbol{N})$-row reduced are recalled in Section 2.

The last condition ensures that A is minimal, in a sense, according to the degree bounds $\boldsymbol{N}$, and that we can easily parametrise which linear combinations of the rows of A are solutions. We recall the relevant definitions and lemmas in Section 2.

We will call such a matrix A a *solution basis*. We will see in Section 2.4 that a solution basis A to a $t \times n$ problem can have at most $t + n$ rows. In the complexities we report here, we cannot afford to compute A explicitly. For example, if all $g_i = x^d$, the number of field elements required to explicitly write down all of the entries of A could be $\Omega((t+n)^2 d)$. Instead, we remark that A is completely given by the problem instance as well as the first t columns of A, containing the λ_j polynomials.[3] Our algorithms will therefore represent A row-wise using the following compact representation.

Definition 1.4. For a given instance of Problem 1.3, a *solution specification* is a tuple $(\boldsymbol{\lambda}, \boldsymbol{\delta}) \in \mathsf{K}[x]^{k \times t} \times \mathbb{Z}_{<0}^{k}$ with $k \leq t+n$ and such that the *completion* of $\boldsymbol{\lambda}$ is a solution basis, and where $\boldsymbol{\delta}$ are the $(-\boldsymbol{N})$-degrees of the rows of A.

The *completion* of $\boldsymbol{\lambda} \in \mathsf{K}[x]^{k \times t}$ with rows $\boldsymbol{\lambda}_j$ is the matrix

$$\left[\begin{array}{c|ccc} \boldsymbol{\lambda}_1 & \operatorname{rem}(\boldsymbol{\lambda}_1 \boldsymbol{S}_1,\ g_1) & \dots & \operatorname{rem}(\boldsymbol{\lambda}_1 \boldsymbol{S}_n,\ g_n) \\ \vdots & & \ddots & \vdots \\ \boldsymbol{\lambda}_k & \operatorname{rem}(\boldsymbol{\lambda}_k \boldsymbol{S}_1,\ g_1) & \dots & \operatorname{rem}(\boldsymbol{\lambda}_k \boldsymbol{S}_n,\ g_n) \end{array}\right] \in \mathsf{K}[x]^{k \times (t+n)} .$$

Note that if $(\boldsymbol{\lambda}, \boldsymbol{\delta})$ is a solution specification, then $\boldsymbol{\delta}$ will consist of only negative numbers, since any solution $\boldsymbol{v}$ by definition has $\deg_{(-\boldsymbol{N})} \boldsymbol{v} < 0$.

Example 1.5. A solution specification for the problem in Example 1.2 is $(\boldsymbol{\lambda}, \boldsymbol{\delta})$ where

$$\boldsymbol{\lambda} = \begin{bmatrix} x^4 + x^3 + x & x^2 + 1 \\ x^3 + x & x \end{bmatrix} \qquad \boldsymbol{\delta} = (-1, -1) .$$

For brevity, here and later, we will sometimes indicate only degrees of matrices: for $F \in \mathsf{K}[x]^{m \times n}$ and $D \in \mathbb{Z}^{m \times n}$, we write $F \trianglelefteq D$ if the degrees of the entries of F are element-wise less than or equal to that of D, with a blank in D representing any negative number (i.e. F has a corresponding 0). The completion of the above solution specification then satisfies

$$A \trianglelefteq \begin{bmatrix} 4 & 2 & 0 & 2 & 3 & 1 \\ 3 & 1 & 0 & 1 & 1 & 3 \end{bmatrix} .$$

One can verify that A is $(-\boldsymbol{N})$-row reduced.

[3] The restriction $N_i \leq \deg g_i$ in Problem 1.1 ensures that for a given $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_t)$, the only possibilities for the ϕ_i in a solution are $\operatorname{rem}(\boldsymbol{\lambda}\boldsymbol{S}_i,\ g_i)$. In particular, if we allowed $N_i > \deg g_i$ then $(0, \dots, 0, g_i, 0, \dots, 0)$ would be a solution which can not be directly reconstructed from its first t elements.

We present two algorithms for solving Problem 1.3, both with complexity $O^{\sim}(n^{\omega-1}td)$ assuming $t < n$, where $d = \max_j T_j + \max_i \deg g_i$ and $O(n^\omega)$ is the cost of multiplying two square polynomial matrices of degree at most d and dimension at most n, see Section 1.1. Both algorithms depend crucially on recent developments on computing minimal approximant basis of matrices with fewer columns than rows [19, 22, 35]. We remark that from the solution basis, one can also compute the expanded form of one or a few of the solutions in the same complexity, for instance if a single, expanded solution to the simultaneous Hermite Padé problem is needed.

Our first algorithm in Section 3 builds on the well-known duality between simultaneous Padé and Hermite Padé which we generalise into a duality theory for minimal approximant basis. If the original problem is $t \times n$ with $t < n$, then the dual will be $n \times t$, and so applying the minimal approximant basis solution recalled in Section 2.4.2 will give a good complexity. Pulling back a solution basis for the dual into a solution for the original requires to efficiently compute t rows of the adjoint of a matrix in Popov form, and this is done by combining partial linearisation [15] and high-order lifting [31].

Our second algorithm works essentially by breaking the $t \times n$ Hermite Padé problem into roughly n/t ones of size roughly $t \times t$: each of these can be solved efficiently using e.g. the minimal approximant basis algorithm. Then, two solution basis can be combined by computing the intersection of their row spaces; this is again handled by a minimal approximant basis computation: a key point here is that we should intersect on only the first t columns of the solution basis, namely those corresponding to λ. A solution basis of the full simultaneous Hermite Padé problem is then obtained by structuring intersections along a binary tree.

This paper is an extension of [29]. In [29] we considered only the simultaneous Padé proble, that is, an input of size $1 \times n$. Here we extend to the general case $t \times n$. In [29] the algorithm based on duality only applied to the case when all g_i where equal to x^d. Here we extend to the general case of arbitrary g_i.

Before we describe our algorithms, we give some preliminary introduction to the main objects and tools employed in Section 2.

Both our algorithms have been implemented in Sage v. 8.3 [30] (though asymptotically slower alternatives to the computational tools are used). The source code can be downloaded from `http://jsrn.dk/code-for-articles`.

Note that our results rely on the not yet published [22], notably for Theorem 2.6. We could instead rely on [20, Theorem 1.4] at the cost of an additional log-factor.

1.1 Cost model

We count basic arithmetic operations in K on an algebraic RAM. We use the following short-hands:

- n^ω is cost of multiplying two square matrices of dimension bounded by n over K.
- $\mathsf{M}(d)$ is the cost of multiplying two polynomials in $\mathsf{K}[x]$ of degree bounded by d.
- $\mathsf{PM}(n, d)$ is the cost of multiplying two square matrices of dimension bounded by n and degree bounded by d.

The currently best known matrix multiplication algorithm has $\omega < 2.373$ [12, 13]. In this paper we will assume $\omega > 2$, otherwise additional log-factors might apply. For example, a nonsingular matrix from $\mathsf{K}^{n\times n}$ can be inverted in time $O(n^\omega)$ field operations from K if $\omega > 2$.

[11] shows $\mathsf{M}(d) \in O(n \log(n) \operatorname{loglog}(n))$, while slightly better results are known for finite fields [17]. We assume $\mathsf{M}(d)$ is super-linear: $\mathsf{M}(d)/d \geq \mathsf{M}(d')/d'$ for all $d \geq d' \geq 1$. We will also assume that there exists an $\epsilon > 0$ such that $\mathsf{M}(d) \in O(n^{\omega-1-\epsilon})$; the purpose of this assumption is to ensure that if fast matrix multiplication techniques are used then fast polynomial multiplication should also be used also. For example, in one of our cost analysis we will use this assumption to make the simplification $\mathsf{M}(d) \log(d)^2 \in O(d^{\omega-1})$.

We will assume $\mathsf{PM}(n, d)$ is super-linear in d: $\mathsf{PM}(n, d) + \mathsf{PM}(n, d') \leq \mathsf{PM}(n, d + d')$ for all $n, d, d' \geq 1$. We will also assume $\mathsf{PM}(n, d) \in \Omega(n^\omega d)$. The currently best known bound over an arbitrary field is given by [11]:

$$\mathsf{PM}(n, d) \in O\big(n^\omega d \log(d) + m^2 d \log(d) \operatorname{loglog}(d))\big) \ .$$

Note that for any positive constants c_1 and c_2 we have $\mathsf{PM}(c_1, d) \in O(\mathsf{M}(d))$ and $\mathsf{PM}(n, c_2) \in O(n^\omega)$. Using an obvious block decomposition and polynomial segmentation we have $\mathsf{PM}(c_1 n, c_2 d) \in O(\mathsf{PM}(n, d))$.

2 Preliminaries

Here we gather together some definitions and results regarding row reduced basis, minimal approximant basis, and their shifted variants. For a matrix A we denote by $A_{i,j}$ the entry in row i and column j. For a matrix A over $\mathsf{K}[x]$ we denote by $\operatorname{Row}(A)$ the $\mathsf{K}[x]$-linear row space of A.

2.1 Degrees and shifted degrees

The degree of a vector $\boldsymbol{v} \in \mathsf{K}[x]^{1\times m}$ or matrix $A \in \mathsf{K}[x]^{n\times m}$ is denoted by $\deg \boldsymbol{v}$ or $\deg A$, and is the maximal degree of entries of $\boldsymbol{v}$ or A (and $\deg 0 := -\infty$). The *row degrees* of A, denoted by $\operatorname{rowdeg} A$, is the tuple $(d_1, \ldots, d_n)$ with $d_i = \deg \operatorname{row}(A, i)$. We similarly introduce *column degrees* denoted $\operatorname{coldeg} A$. When we compare tuples of integers, e.g. $\operatorname{rowdeg} A_1 < \operatorname{rowdeg} A_2$, we mean that the comparison holds element-wise.

The (row-wise) *leading matrix* of A, denoted by $\mathrm{LM}(A) \in \mathsf{K}^{n\times m}$, has $\mathrm{LM}(A)_{i,j}$ equal to the coefficient of x^{d_i} of $A_{i,j}$.

Next we recall [2, 19, 35] the shifted variants of the notion of degree, row degrees, and leading matrix. For a *shift* $\boldsymbol{s} = (s_1, \ldots, s_n) \in \mathbb{Z}^m$, define the $m \times m$ diagonal matrix $x^{\boldsymbol{s}}$ by

$$x^{\boldsymbol{s}} := \begin{bmatrix} x^{s_1} & & \\ & \ddots & \\ & & x^{s_m} \end{bmatrix}.$$

Then the *$\boldsymbol{s}$-degree* of v, the *$\boldsymbol{s}$-row degrees* of A, and the *$\boldsymbol{s}$-leading matrix* of A, are defined by $\deg_{\boldsymbol{s}} v := \deg vx^{\boldsymbol{s}}$, $\mathrm{rowdeg}_{\boldsymbol{s}} A := \mathrm{rowdeg}\, Ax^{\boldsymbol{s}}$, and $\mathrm{LM}_{\boldsymbol{s}}(A) := \mathrm{LM}(Ax^{\boldsymbol{s}})$. For a shift $t \in \mathbb{Z}^n$ we similarly have the *$\boldsymbol{t}$*-column degree $\mathrm{coldeg}_{\boldsymbol{t}} A := \mathrm{coldeg}\, x^{\boldsymbol{t}} A = \mathrm{rowdeg}_{\boldsymbol{t}} A^{\top}$. Note that with negative entries in $\boldsymbol{s}$, we pass over the ring of Laurent polynomials only for convenience; our algorithms will only compute with polynomials. As pointed out in [19], up to negation the definition of $\boldsymbol{s}$-degree is equivalent to that used in [8] and to the notion of *defect* in [5].

For a vector $\boldsymbol{v} \in \mathsf{K}[x]^{1\times k}$, we denote by $\mathrm{diag}(\boldsymbol{v})$ the diagonal matrix with the entries of $\boldsymbol{v}$.

2.2 Row and column reduced

Although row reducedness can be defined for matrices of arbitrary shape and rank, it suffices here to consider the case of matrices of full row rank. A matrix $R \in \mathsf{K}[x]^{n\times m}$ of full row rank n is *$\boldsymbol{s}$-row reduced* if any of the equivalent conditions in the following theorem are satisfied. If all entries in the shift $\boldsymbol{s} \in \mathbb{Z}^m$ are identical we simply say R is *row reduced*.

Theorem 2.1 (see [24, Section 6.3.2])**.** *Let $R \in \mathsf{K}[x]^{n\times m}$ have full row rank n and let $\boldsymbol{s} \in \mathbb{Z}^m$ be a shift. Then the following are equivalent:*

1. *$\mathrm{LM}_{\boldsymbol{s}}(R)$ has full row rank n.*
2. *Among all matrices that are left equivalent to R, the list of $\boldsymbol{s}$-degrees of the rows of R, when sorted in non-decreasing order, will be lexicographically minimal.*
3. *For any $\boldsymbol{v} \in \mathsf{K}[x]^{1\times n}$, we have*

$$\deg_{\boldsymbol{s}}(\boldsymbol{v}R) = \max_{i=1,\ldots,n} \left(\deg_{\boldsymbol{s}} \mathrm{row}(R, i) + \deg v_i\right).$$

Property 3 in Theorem 2.1 is known as the "predictable degree"-property [24, Theorem 6.3-13]. Every $A \in \mathsf{K}[x]^{n\times m}$ of full row rank is left equivalent to a matrix $R \in \mathsf{K}[x]^{n\times m}$ that is $\boldsymbol{s}$-row reduced. The notion of row reducedness

has a column-wise counterpart: a matrix $R \in \mathsf{K}[x]^{m\times n}$ is *column reduced* if $R^\top$ is row reduced, and $\boldsymbol{s}$-column reduced if $R^\top$ is $\boldsymbol{s}$-row reduced. We will mostly be working with row reduced, and the LM-notation applies to this, but in some instances we will use column reduced to simplify notation.

The following is a well-known fact on row-reduced matrices:

Lemma 2.2. *Let $F_1 \in \mathsf{K}[x]^{m\times n}$ over $\mathsf{K}[x]$ be $\boldsymbol{s}$-row reduced, and $F_2 \in \mathsf{K}[x]^{k\times m}$ be $\boldsymbol{r}$-row reduced where $\boldsymbol{r} = \operatorname{rowdeg}_{\boldsymbol{s}} F_1$. Then F_2F_1 is $\boldsymbol{s}$-row reduced with $\operatorname{rowdeg}_{\boldsymbol{s}}(F_2F_1) = \operatorname{rowdeg}_{\boldsymbol{r}}(F_2)$.*

Proof. Note that since F_1 and F_2 are shifted row reduced, they have full row rank and so $k \le m \le n$. By applying the predictable degree property for each row of F_2, we immediately get:

$$\operatorname{rowdeg}_{\boldsymbol{s}}(F_2F_1) = \operatorname{rowdeg} F_2 + \operatorname{rowdeg}_{\boldsymbol{s}} F_1 = \operatorname{rowdeg}_{\boldsymbol{r}} F_2.$$

We have left to show that $\operatorname{LM}_{\boldsymbol{s}}(F_2F_1)$ has full row rank. Since $\operatorname{LM}_{\boldsymbol{s}}(F_2F_1) = \operatorname{LM}(x^{\boldsymbol{t}}(F_2F_1)x^{\boldsymbol{s}})$ for any shift $\boldsymbol{t}$, we consider

$$x^{-\operatorname{rowdeg}_{\boldsymbol{r}} F_2} F_2F_1x^{\boldsymbol{s}} = (x^{-\operatorname{rowdeg}_{\boldsymbol{r}} F_2} F_2x^{\boldsymbol{r}})(x^{-\boldsymbol{r}}F_1x^{\boldsymbol{s}}) =: AB\ .$$

Now both A and B are row reduced with row degrees all 0. Thus

$$AB = \big(A_0x^0 + O(x^{-1})\big)\big(B_0x^0 + O(x^{-1})\big) = A_0B_0 + O(x^{-1})\ ,$$

where A_0 and B_0 are over K and have full row rank, whereby A_0B_0 has full row rank. Therefore AB is row reduced, i.e., F_2F_1 is $\boldsymbol{s}$-row reduced. □

A canonical $\boldsymbol{s}$-row reduced basis is provided by the (row-wise) $\boldsymbol{s}$-Popov form. Although an $\boldsymbol{s}$-Popov form can be defined for a matrix of arbitrary shape and rank, it suffices here to consider the case of a non-singular matrix. The following definition is equivalent to [19, Definition 1.2].

Definition 2.3. A non-singular matrix $R \in \mathsf{K}[x]^{n\times n}$ is in $\boldsymbol{s}$-Popov form if $\operatorname{LM}_{\boldsymbol{s}}(R)$ is unit lower triangular and the degrees of off-diagonal entries of R are strictly less than the degree of the diagonal entry in the same column.

A matrix R is in *column* $\boldsymbol{s}$-Popov form if $R^\top$ is in $\boldsymbol{s}$-Popov form.

2.3 Minimal approximant basis

We recall the standard notion of (left) minimal approximant basis, sometimes known as order basis or σ-basis [5]. For a matrix $A \in \mathsf{K}[x]^{n\times m}$ and order $d \in \mathbb{Z}_{\ge 0}$, an *order d (left) approximant* is a vector $\boldsymbol{p} \in \mathsf{K}[x]^{1\times n}$ such that $\boldsymbol{p}A \equiv \boldsymbol{0} \mod x^d$.

A (left) *approximant basis of order d* is then a matrix $F \in \mathsf{K}[x]^{n\times n}$ which is a basis of all order d approximants. Such a basis always exists and has

full rank n. For a shift $\boldsymbol{s} \in \mathbb{Z}^n$, F is then an *$\boldsymbol{s}$-minimal approximant basis* if it is $\boldsymbol{s}$-row reduced.

We will also consider right approximants, i.e. a vector $\boldsymbol{p} \in \mathsf{K}[x]^{m \times 1}$ such that $A\boldsymbol{p} \equiv \boldsymbol{0} \mod x^d$, as well as the related notions of right approximant basis and right minimal approximant basis. When we omit the direction, we mean left approximant.

Let $\mathsf{MinBasis}(d, A, \boldsymbol{s})$ be a function that returns $(F, \boldsymbol{\delta})$, where F is an $\boldsymbol{s}$-minimal left approximant basis of A of order d, and $\boldsymbol{\delta} = \operatorname{rowdeg}_{\boldsymbol{s}} F$. Note that F is not canonical, and we allow $\mathsf{MinBasis}$ to return any such $\boldsymbol{s}$-minimal approximant basis. It follows from standard properties of row reducedness that $\boldsymbol{\delta}$ will be the same for all of these.

The next lemma recalls a well known method of constructing minimal approximant basis recursively. We stress again that although the output of $\mathsf{MinBasis}$ is not unique, the lemma holds for *any* $\boldsymbol{s}$-minimal approximant basis that $\mathsf{MinBasis}$ might return.

Lemma 2.4. *Let* $A = \left[\begin{array}{c|c} A_1 & A_2 \end{array}\right]$ *over* $\mathsf{K}[x]$. *If* $(F_1, \boldsymbol{\delta}_1) = \mathsf{MinBasis}(d, A_1, \boldsymbol{s})$ *and* $(F_2, \boldsymbol{\delta}_2) = \mathsf{MinBasis}(d, F_1 A_2, \boldsymbol{\delta}_1)$, *then* $F_2 F_1$ *is an* $\boldsymbol{s}$*-minimal approximant basis of* A *of order* d *with* $\boldsymbol{\delta}_2 = \operatorname{rowdeg}_{\boldsymbol{s}} F_2 F_1$.

Sometimes only the *negative part* of an $\boldsymbol{s}$-minimal approximant basis is required, i.e. the submatrix of the approximant basis consisting of rows with negative $\boldsymbol{s}$-degree. Let function $\mathsf{NegMinBasis}(d, A, \boldsymbol{s})$ have the same output as $\mathsf{MinBasis}$, but with F restricted to the negative part.

Corollary 2.5. *Lemma 2.4 still holds if* $\mathsf{MinBasis}$ *is replaced by* $\mathsf{NegMinBasis}$, *and "an* $\boldsymbol{s}$*-minimal" is replaced with "the negative part of an* $\boldsymbol{s}$*-minimal."*

An approximant basis always has a determinant which is a power of x. Moreover, if F is a minimal approximant basis of order d for an $A \in \mathsf{K}[x]^{n \times m}$, then $\det F \mid x^{dm}$. To see this, note that there exists a unimodular matrix $U \in \mathsf{K}[x]^{n \times n}$ such that that UA has last $n - m$ rows zero. Then the matrix $\bar{U}$ obtained from U by multiplying the first m rows by x^d will satisfy $\bar{U}A \equiv 0 \mod x^d$, so the row space of $\bar{U}$, which has determinant x^{md} up to a constant, must be contained in the row space of any approximant basis F of A.

Many problems of $\mathsf{K}[x]$ matrices or approximations reduce to the computation of (shifted) minimal approximant basis, see e.g. [5, 14], often resulting in the best known asymptotic complexities for these problems. Part 1 of the following theorem is a special case of [22, Theorem 1.1]. Part 2 is [21, Proposition 7.1].

Theorem 2.6. *There exists an algorithm* $\mathsf{PopovMinBasis}(d, A, \boldsymbol{s})$ *implementing* $\mathsf{MinBasis}$ *and such that the minimal approximant basis is in* $\boldsymbol{s}$*-Popov form. Assume* $A \in \mathsf{K}[x]^{n \times m}$ *satisfies* $m \leq n$ *and* $\deg A \leq d$. *In terms of operations from* K, *then* $\mathsf{PopovMinBasis}(d, A, \boldsymbol{s})$ *has cost bounded by*

1. $O(\mathsf{PM}(n, md/n)\log(md/n)^2 + n^{\omega-1}md\log(n)))$.

2. $O(n(md)^{\omega-1} + (md)^{\omega}\log(d))$ *if* $md \in O(n)$.

We will also use $\mathsf{PopovMinBasis}_{\mathsf{Right}}$ to the transpose of PopovMinBasis, which computes a right minimal approximant basis in shifted column-Popov form.

Note that [22] contains improvements of the above on the level of logarithmic factors for various special cases; however, none of these can straightforwardly be applied to our case.

At the time of writing, [22] is not yet published. We could instead rely on the earlier [19, Theorem 1.4], which would instead have the cost $O(n^{\omega-1}dm\log(dm)^2\log(dm/n)^2\operatorname{loglog}(dm))$.

2.4 Existing algorithms for simultaneous Hermite Padé approximations

Let $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$ be an instance of Problem 1.3 of size $t \times n$. We recall two known approaches for computing a solution specification using row reduction and minimal approximant basis computation. We will discuss the latter in greater detail since we will build upon it for our algorithm in Section 3.

2.4.1 Via reduced basis

Using the predictable degree property it is easy to show that if $R \in \mathsf{K}[x]^{(n+t)\times(n+t)}$ is an $(-\boldsymbol{N})$-row reduced basis of

$$A = \left[\begin{array}{c|c} I_{t\times t} & \boldsymbol{S} \\ \hline & \operatorname{diag}(\boldsymbol{g}) \end{array}\right] \in \mathsf{K}[x]^{(n+t)\times(n+t)}, \tag{2.1}$$

then the sub-matrix of R comprised of the rows with negative $(-\boldsymbol{N})$-degree form a solution basis. Therefore, if $\boldsymbol{\lambda}$ are the first t columns of this sub-matrix and $\boldsymbol{\delta}$ the $(-\boldsymbol{N})$-row degrees of the sub-matrix, then $(\boldsymbol{\lambda}, \boldsymbol{\delta})$ forms a solution specification. This shows why a solution specification has at most $t + n$ entries. If $d = \deg \boldsymbol{g}$ then one can use [26] to compute the $(-\boldsymbol{N})$-shifted Popov form of A at the cost $O^{\sim}(n(n+t)^{\omega-1}d)$.

When $t = n = 1$, the extended Euclidean algorithm on input $S_{1,1}$ and g_1 can solve the approximation problem by essentially computing a row reduced basis of the 2×2 matrix A: each iteration corresponds to a reduced basis for a range of possible shifts [16, 23, 32]. The complexity of this is $O(\mathsf{M}(\deg g_1)\log(\deg g_1))$.

2.4.2 Via minimal approximant basis

Consider the special case when $\boldsymbol{g} = (x^d, x^d, \ldots x^d)$, that is, all $g_i = x^d$ for a common d. An approximant $\boldsymbol{v} = (\boldsymbol{\lambda} \mid \phi_1, \ldots, \phi_n) \in \mathsf{K}[x]^{t+n}$ of order d of

$$B = \begin{bmatrix} -\boldsymbol{S} \\ I_{n\times n} \end{bmatrix} \in \mathsf{K}[x]^{(n+t)\times n}$$

clearly satisfies $\boldsymbol{\lambda}\boldsymbol{S}_i \equiv \phi_i \mod x^d$ for $i = 1, \ldots, n$; conversely, any such vector $\boldsymbol{v}$ satisfying these congruences must be an approximant of B of order d. So the negative part of a $(-\boldsymbol{N})$-minimal approximant basis of B of order d is a solution basis to the simultaneous Hermite Padé approximation.

In the case of arbitrary $\boldsymbol{g}$ we can reduce to computing a minimal approximant basis of the augmented input

$$B = \begin{bmatrix} -\boldsymbol{S} \\ I_{n\times n} \\ \operatorname{diag}(\boldsymbol{g}) \end{bmatrix} \in \mathsf{K}[x]^{(2n+t)\times n}. \tag{2.2}$$

To understand the approach, note that a left kernel basis for B in (2.2) is given by

$$K = \left[\begin{array}{c|c} A & * \end{array}\right] = \left[\begin{array}{cc|c} I_{t\times t} & \boldsymbol{S} & \\ & \operatorname{diag}(\boldsymbol{g}) & -I_{n\times n} \end{array}\right],$$

with the principal submatrix $A \in \mathsf{K}[x]^{(n+t)\times(n+t)}$ of K is the lattice in (2.1). The rows with negative $(-\boldsymbol{N})$-degree in a reduced basis for A give a solution basis to the problem instance. For a well chosen shift $\boldsymbol{h}$ and order d, the negative part of an $\boldsymbol{h}$-minimal approximant basis of order d of B will contain the negative part of a $(-\boldsymbol{N})$-row reduced basis of A. Algorithm 1 formalises this, and its correctness and choice of shift $\boldsymbol{h}$ and order d is due to the following result.

Theorem 2.7. *Corresponding to an instance $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$ of Problem 1.3 of size $t \times n$, define a shift $\boldsymbol{h}$ and order d:*

- $\boldsymbol{h} := -(\boldsymbol{N} \mid T-1, \ldots, T-1) \in \mathbb{Z}^{2n+t}$, *where* $T = \max_j\{T_j\}$
- $d := T + \max_i \deg g_i - 1$

If $(G, \boldsymbol{\delta}) = \mathsf{NegMinBasis}(d, B, \boldsymbol{h})$ where

$$B = \begin{bmatrix} -\boldsymbol{S} \\ I_{n\times n} \\ \operatorname{diag}(\boldsymbol{g}) \end{bmatrix} \in \mathsf{K}[x]^{(2n+t)\times n},$$

then the submatrix of G comprised of the first $n+t$ columns is a solution basis to the problem instance.

Proof. The left kernel of B consists of exactly those vectors $\boldsymbol{v} = (\boldsymbol{\lambda} \mid \phi_1, \ldots, \phi_n, q_1, \ldots, q_n)$ such that

$$\boldsymbol{\lambda} \boldsymbol{S}_i = \phi_i + q_i g_i \ .$$

If such a vector $\boldsymbol{v}$ has $\deg_{\boldsymbol{h}} \boldsymbol{v} < 0$, then $\boldsymbol{v}' = (\boldsymbol{\lambda} \mid \phi_1, \ldots, \phi_n)$ is a solution to the simultaneous Hermite Padé approximation problem.

Conversely, any solution $\boldsymbol{v}' = (\boldsymbol{\lambda} \mid \phi_1, \ldots, \phi_n)$ with $\deg_{(-\boldsymbol{N})} \boldsymbol{v}' < 0$ can be extended to $\boldsymbol{v} = (\boldsymbol{v}' \mid q_1, \ldots, q_n)$ such that the above equality holds: since $\deg \phi_i < \deg g_i$ we must have q_i equal to the quotient of $\boldsymbol{\lambda} \boldsymbol{S}_i$ divided by g_i, $1 \leq i \leq n$. By the definition of shifted degree, we have

$$\deg_{\boldsymbol{h}} \boldsymbol{v} = \max(\deg_{(-\boldsymbol{N})} \boldsymbol{v}', \deg_{-(T-1,\ldots,T-1)}[q_1, \ldots, q_n]).$$

We claim that $\deg_{-(T-1,\ldots,T-1)}[q_1, \ldots, q_n] \leq \deg_{(-\boldsymbol{N})} \boldsymbol{v}'$, so that $\deg_{\boldsymbol{h}} \boldsymbol{v} = \deg_{(-\boldsymbol{N})} \boldsymbol{v}' < 0$. To see this, note that

$$\begin{aligned} \deg q_i &= \deg \boldsymbol{\lambda} \boldsymbol{S}_i - \deg g_i \\ &\leq \overbrace{(\max_j T_j + \deg_{(-\boldsymbol{N})} \boldsymbol{v}')}^{\geq \deg \boldsymbol{\lambda}} + \overbrace{\deg g_i - 1}^{\geq \deg \boldsymbol{S}_i} - \deg g_i \\ &= T + \deg_{(-\boldsymbol{N})} \boldsymbol{v}' - 1. \end{aligned}$$

Thus solutions to the simultaneous Hermite Padé approximation problems correspond exactly to vectors in the left kernel space of B with negative $\boldsymbol{h}$-degree. We claim that the set of such kernel vectors are exactly the approximants of B of order d of negative $\boldsymbol{h}$-degree: That such vectors in the left kernel are approximants is obvious. Consider now a minimal approximant of B of order d, $\boldsymbol{v} = (\boldsymbol{\lambda} \mid \phi_1, \ldots, \phi_n, q_1, \ldots, q_n)$ with $\deg_{\boldsymbol{h}} \boldsymbol{v} < 0$. By the shape of B, then $\boldsymbol{\lambda} \boldsymbol{S}_i \equiv \phi_i + q_i g_i \mod x^d$ for $i = 1, \ldots, n$. But all terms in the congruence must have degree strictly less than d, and thus the congruence lifts to an equality. Therefore $\boldsymbol{v}$ is in the left kernel of B.

Thus G spans all the left kernel vectors of negative $\boldsymbol{h}$-degree, and the submatrix G' comprised of the first $n + t$ columns of G therefore spans all solutions to the simultaneous Hermite Padé approximation. G' is therefore a solution basis if it is $(-\boldsymbol{N})$-row reduced. But this follows from part 2 of Theorem 2.1 because $\operatorname{rowdeg}_{\boldsymbol{h}} G = \operatorname{rowdeg}_{(-\boldsymbol{N})} G'$ and G is $\boldsymbol{h}$-row reduced. □

From Theorem 2.6 we get:

Corollary 2.8. *Let $d = \max T_i + \max \deg g_i$. In terms of operations from* K, DirectSHPade *has cost bounded by*

1. $O\big(\mathsf{PM}(n+t, \frac{nd}{n+t}) \log(\frac{nd}{n+t})^2 + (n+t)^{\omega-1} nd \log(n+t)\big)$.
2. $O((n+t)(nd)^{\omega-1} + (nd)^{\omega} \log(d))$ *if* $nd \in O(n+t)$.

Algorithm 1 DirectSHPade

Input: $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$, an instance of Problem 1.3 of size $t \times n$.
Output: $(\boldsymbol{\lambda}, \boldsymbol{\delta})$, a solution specification.

1 $\boldsymbol{h} \leftarrow -(\boldsymbol{N} \mid T-1, \ldots, T-1) \in \mathbb{Z}^{2n+t}$, where $T = \max_i T_i$
2 $d \leftarrow T + \max_i \deg g_i - 1$
3 $B = \begin{bmatrix} -\boldsymbol{S} \\ I_{n\times n} \\ \operatorname{diag}(\boldsymbol{g}) \end{bmatrix}$
4 $(\left[\begin{array}{c|c} \boldsymbol{\lambda} & * \end{array}\right], \boldsymbol{\delta}) \leftarrow \mathsf{NegMinBasis}(d, B, \boldsymbol{h})$
5 **return** $(\boldsymbol{\lambda}, \boldsymbol{\delta})$

Note that in the case $t \geq n$ then the above is the desirable $O^{\sim}\big(nt^{\omega-1}d\big)$. However when $t < n$ — the case that we focus on in this paper — this approach simply gives $O^{\sim}\big(n^{\omega}d\big)$.

3 Algorithm 1: Reduction to the Dual

In this section we present our first algorithms for solving the simultaneous Hermite Padé problem. The algorithm essentially proceeds as DirectSHPade and computes a minimal approximant basis of the following matrix:

$$\hat{B} = \left[\begin{array}{c|c|c} x^d I_{t\times t} & -\boldsymbol{S} & \\ \hline & I_{n\times n} & \\ \hline & \operatorname{diag}(\boldsymbol{g}) & x^d I_{n\times n} \end{array}\right] \in \mathsf{K}[x]^{(2n+t)\times(2n+t)}$$

To optimally leverage the efficient minimal approximant basis computation of Theorem 2.6, we first compute a right minimal approximant basis of $x^d\hat{B}^{-1} \in \mathsf{K}[x]^{(2n+t)\times(2n+t)}$ and then compute a solution basis from that. This approach is reminiscent of the well-known duality between the simultaneous Padé problem and the Hermite Padé problem: this duality, first observed in a special case [25], and then later more generally [4, 6], was previously exploited in [7] to develop algorithms for the fraction-free computation of simultaneous Padé approximations.

We first develop a general theory of minimal approximant basis "duality", and how to perform the computations efficiently. We then apply this without further ado to the matrix B.

3.1 Duals of minimal approximant basis

For a nonsingular $n \times n$ matrix A recall that the adjoint of A, denoted by $\operatorname{adj}(A)$, is equal to $(\det A)A^{-1}$, and that entry $\operatorname{adj}(A)_{j,i}$ is equal to $(-1)^{i+j}$ times the determinant of the $(n-1)\times(n-1)$ submatrix that is obtained from A by deleting row i and column j. In particular, $\operatorname{adj}(A)$ is a polynomial matrix.

Lemma 3.1. *Let $A \in \mathsf{K}[x]^{n\times n}$ be $\boldsymbol{s}$-row reduced. Then* $\operatorname{adj}(A)$ *is $(-\boldsymbol{s})$-column reduced with*

$$\operatorname{coldeg}_{(-\boldsymbol{s})}\operatorname{adj}(A) = (d-\eta_1, \ldots, d-\eta_n) \ ,$$

where $\boldsymbol{\eta} = \operatorname{rowdeg}_{\boldsymbol{s}} A$ and $d = \deg\det A = \sum_i (\eta_i - s_i)$.

Proof. Since A is $\boldsymbol{s}$-row reduced then $Ax^{\boldsymbol{s}}$ is row reduced. Note that $(Ax^{\boldsymbol{s}})\operatorname{adj}(Ax^{\boldsymbol{s}}) = (\det Ax^{\boldsymbol{s}})I_m$. Let $\eta := \sum_i \eta_i = \deg\det Ax^{\boldsymbol{s}}$. It follows that column i of $\operatorname{adj}(Ax^{\boldsymbol{s}})$ must have degree at least $\eta - \eta_i$ since η_i is the degree of row i of $(Ax^{\boldsymbol{s}})$. However, entries in column i of $\operatorname{adj}(Ax^{\boldsymbol{s}})$ are minors of the matrix obtained from $Ax^{\boldsymbol{s}}$ by removing row i, hence have degree at most $\eta - \eta_i$. It follows that the column-wise leading coefficient matrix of $\operatorname{adj}(Ax^{\boldsymbol{s}})$ is nonsingular, hence $\operatorname{adj}(Ax^{\boldsymbol{s}})$ is column reduced. Since $\operatorname{adj}(Ax^{\boldsymbol{s}}) = (\det x^{\boldsymbol{s}})x^{-\boldsymbol{s}}\operatorname{adj}(A)$ we conclude that $\operatorname{adj}(A)$ is $(-\boldsymbol{s})$-column reduced with $\operatorname{coldeg}_{(-\boldsymbol{s})}\operatorname{adj}(A) = (\eta - \eta_1 - s, \ldots, \eta - \eta_n - s)$. □

For any $\boldsymbol{s}$-row reduced matrix, Lemma 3.1 defines, via the adjoint, a unique $(-\boldsymbol{s})$-column reduced dual. Our goal is to establish a similar duality for minimal approximant basis. We begin with the following result.

Lemma 3.2. *Let $A, B \in \mathsf{K}[x]^{n\times n}$ such that $AB = x^d I_n$. Then A is a left approximant basis for B of order d, and B is a right approximant basis for A of order d.*

Proof. Let G be any approximant basis for A of order d. Then $AG = x^d R$ for some $R \in \mathsf{K}[x]^{n\times n}$. Let $k \leq nd$ be such that $\det B$ is an associate of x^k. Clearly, the columns of B are right approximants of A of order d, so $\det G$ divides $\det B$. But $G = A^{-1}Rx^d = BR$ so $\det G = (\det B)(\det R)$. It follows that $\det R$ has degree zero, so $\det G$ is an associate of x^k and B is a right approximant basis. By symmetry, A is a left approximant basis for B of order d. □

Lemma 3.3. *If $A \in \mathsf{K}[x]^{n\times n}$ is a left approximant basis of order d for some input matrix in $\mathsf{K}[x]^{n\times *}$, then $x^d A^{-1}$ is a polynomial matrix. Similarly, if $B \in \mathsf{K}[x]^{n\times n}$ is a right approximant basis for some input matrix in $\mathsf{K}[x]^{*\times n}$, then $x^d B^{-1}$ is a polynomial matrix.*

Proof. The rows of $x^d I_n$ are all approximants of order d, so they are contained in the row space of A. Therefore there is an $B \in \mathsf{K}[x]^{n\times n}$ such that $BA = x^d I_n$, and hence $x^d A^{-1} = B$. The second claim is symmetric. □

The duality of Lemma 3.2 thus holds in general. That is, if A is as in Lemma 3.3, then $B = x^d A^{-1}$ is over $\mathsf{K}[x]$ with

$$AB = x^d I_n \qquad B = x^d A^{-1} \qquad A = x^d B^{-1}. \tag{3.1}$$

Symmetrically, if B is as in Lemma 3.3, then $A = x^d B$ is over $\mathsf{K}[x]$ and (3.1) also holds. Every left (right) minimal approximant basis thus has a natural right (left) dual basis.

Proposition 3.4. *Let $A, B \in \mathsf{K}[x]^{n\times n}$ such that $AB = x^d I_n$. If G is a right $\boldsymbol{s}$-minimal approximant basis for A of order d, then $x^d G^{-1}$ is a left $(-\boldsymbol{s})$-minimal approximant basis for B of order d. Also, if $\operatorname{coldeg}_{\boldsymbol{s}} G = (\eta_1, \ldots, \eta_n)$, then $\operatorname{rowdeg}_{(-\boldsymbol{s})}(x^d G^{-1}) = (d - \eta_1, \ldots, d - \eta_n)$.*

Proof. The proof of Lemma 3.3 established that

- $AG = x^d R$ for an $R \in \mathsf{K}[x]^{n\times n}$ with $\deg\det R = 0$, and
- that if $\det B$ is an associate of x^k, then $\det G$ is an associate of x^k also.

By Lemma 3.3, $x^d G^{-1}$ is a polynomial matrix. Write now

$$x^d I_n = AB = AGG^{-1}B = Rx^d G^{-1}B$$

Hence, $(x^d G^{-1})B = x^d R^{-1}$ where $R^{-1} \in \mathsf{K}[x]^{n\times n}$ since $\deg\det R = 0$, and so each row of $x^d G^{-1}$ is a left approximant for B of order d. By Lemma 3.3 A is a left approximant basis for B of order d. But then since $\det(x^d G^{-1})$ is an associate of $\det A$, then $x^d G^{-1}$ must be a left approximant basis for B of order d.

Next we show that $x^d G^{-1}$ is $(-\boldsymbol{s})$-row reduced. Since G is $\boldsymbol{s}$-column reduced, by Lemma 3.1 $\operatorname{adj}(G)$ is $(-\boldsymbol{s})$-row reduced with

$$\operatorname{rowdeg}_{(-\boldsymbol{s})}\operatorname{adj}(G) = (k - \eta_1, \ldots, k - \eta_n),$$

where $(\eta_1, \ldots, \eta_n) = \operatorname{coldeg}_{\boldsymbol{s}} G$. Since $\operatorname{adj}(G) = x^{k-d}(x^d G^{-1})$ then $x^d G^{-1}$ must also be $(-\boldsymbol{s})$-row reduced with

$$\operatorname{rowdeg}_{(-\boldsymbol{s})}(x^d G^{-1}) = (d - \eta_1, \ldots, d - \eta_n).$$

□

Suppose a nonsingular $B \in \mathsf{K}[x]^{n\times n}$ enjoys the property that $x^d B^{-1}$ is over $\mathsf{K}[x]$. Then Proposition 3.4 gives the following recipe to compute a left minimal approximant basis F of order d for B.

1. Compute $A = x^d B^{-1}$.
2. Compute a right minimal approximant basis G of order d for A.
3. Compute $F = x^d G^{-1}$.

Applying the above recipe can, for some inputs, reduce to a minimal approximant basis problem of smaller dimension. For example, if $\boldsymbol{S} \in \mathsf{K}[x]^{1\times n}$ and

$$B = \left[\begin{array}{c|c} x^d & -\boldsymbol{S} \\ \hline & I_{n\times n} \end{array}\right] \in \mathsf{K}[x]^{(n+1)\times(n+1)},$$

then

$$A = x^d B^{-1} = \left[\begin{array}{c|c} 1 & \boldsymbol{S} \\ \hline & x^d I_{n\times n} \end{array}\right] \in \mathsf{K}[x]^{(n+1)\times(n+1)}.$$

Clearly, a right minimal approximant basis for just the first row of A will be a right minimal approximant basis for the entire matrix A.

In the following two sections, we detail how the above recipe can be leveraged efficiently for simultaneous Hermite Padé problems.

3.2 Computing only part of the dual

Here we show how to compute the first m rows of the inverse of $F := \mathsf{PopovMinBasis}(d, A, \boldsymbol{s})$ in about the same time as the cost bound given by Theorem 2.6 to compute F.

Theorem 3.5. *Let $F \in \mathsf{K}[x]^{n\times n}$ be a minimal approximant basis of order d, in shifted Popov form, for an input matrix $A \in \mathsf{K}[x]^{n\times m}$ with $m \leq n$. Then In terms of operations from K, the first m rows of $x^d F^{-1}$ can be computed in time*

1. $O(\log(n/m)(\mathsf{PM}(n, md/n) + nm\,\mathsf{M}(d)))$.
2. $O(\log(d)(md)^\omega)$ *if* $md < n$.

Our result relies on many properties enjoyed by F, which we summarize in the following lemma.

Lemma 3.6. *Let $F \in \mathsf{K}[x]^{n\times n}$ be as in Theorem 3.5. Then*

1. $\det F$ *is a divisor of* x^{md}.
2. $\sum \mathrm{coldeg} F \leq md$.
3. $\deg F^{-1} \leq 0$.
4. $x^d F^{-1}$ *is over* $\mathsf{K}[x]$.
5. $\deg x^d F^{-1} \leq d$.

Proof. Since F is a minimal approximant basis, $\det F$ will be a power of x. Since F is a minimal approximant basis of order d for an input with m columns, $\deg\det F \leq md$. Part 1 follows from these observations.

Part 2 follows from part 1 because $\sum \mathrm{coldeg} F = \deg\det F$ since F is in shifted Popov form.

Part 3 holds because any matrix in shifted Popov form is column reduced.
Part 4 is Lemma 3.3.
Part 5 follows from parts 3 and 4. □

Computing the first m rows of F^{-1} is equivalent to solving the following nonsingular linear system:

$$\left[\begin{array}{c|c} I_{m\times m} & 0_{m\times(n-m)} \end{array}\right] F^{-1}.$$

High-order lifting [31, Algorithm 6] gives a reduction of linear system solving to matrix multiplication. The cost of high-order lifting is sensitive to $\deg F$. To avoid a cost blowup because of potentially skewed column degrees, we first use partial linearisation to transform our linear system solving problem to one involving a matrix with degree bounded by the *average* column degree of F. The next result follow from [15, Corollary 2].

Lemma 3.7. *Let $F \in \mathsf{K}[x]^{n\times n}$ be as in Theorem 3.5. It is possible to construct from F, with no operations from K, a matrix $G \in \mathsf{K}[x]^{\bar{n}\times\bar{n}}$ with*

$$G^{-1} = \left[\begin{array}{c|c} F^{-1} & * \\ \hline * & * \end{array}\right],$$

and such that G enjoys the following properties:

- $\deg G \leq \lceil md/n \rceil$.
- $n \leq \bar{n} < 2n$
- $\det G = \det F$

Then the first n columns of

$$\left[\begin{array}{c|c} x^d I_{m\times m} & 0_{m\times(\bar{n}-m)} \end{array}\right] G^{-1} \in \mathsf{K}[x]^{m\times\bar{n}} \tag{3.2}$$

will be the first m rows of $x^d F^{-1}$, and since $x^d F^{-1} \in \mathsf{K}[x]$ and $\deg x^d F^{-1} \leq d$ (Lemma 3.6 parts 4 and 5) these first n columns will be over $\mathsf{K}[x]$ with degree bounded by d. The next lemma establishes the first part of Theorem 3.5.

Lemma 3.8. *Let $F \in \mathsf{K}[x]^{n\times n}$ be as in Theorem 3.5. If $md \geq n$ then the first m rows of F^{-1} can be computed in*

$$O(\log(n/m)(\mathsf{PM}(n, md/n) + nm\,\mathsf{M}(d)))$$

field operations in K.

Proof. We will compute the system solution (3.2) using high-order lifting. This requires a modulus X that is relatively prime to $\det G$, and with $\deg X \geq \deg G$. Since $\det G$ is a power of x (Lemma 3.6.1), and the linear polynomial $x - 1$ exists over any field, we can set $X := (x-1)^{\lceil md/n \rceil}$. As an

initialization, high-order lifting requires the inverse of G modulo X. This is computed in time $O(\mathsf{PM}(\bar{n}, \deg X))$ by first computing the inverse of the scalar matrix $G \bmod (x-1) \in \mathsf{K}^{\bar{n}\times\bar{n}}$ and then using quadratic Newton iteration to get $G^{-1} \bmod X$. Since $\bar{n} < 2n$ and $\deg X \leq 1 + md/n$ we have $\mathsf{PM}(\bar{n}, \deg X) \in O(\mathsf{PM}(n, md/n)$.

High-order lifting will compute the X-adic series expanion of (3.2) to a desired precision $p \in \mathbb{Z}_{\geq 0}$. Since $\deg x^d F^{-1} \leq d$ (Lemma 3.6.5), we require to lift up to X^p for a p with $\deg X^p > d$: the minimal such p is $p := 1 + \lfloor d/\deg X \rfloor$. By [31, Proposition 15] the lifting then has cost

$$O\big((\log \bar{p})\, \lceil m\bar{p}/\bar{n} \rceil\, \mathsf{PM}(\bar{n}, \deg X))\big) \tag{3.3}$$

operations in K, where $\bar{p} < 2p$ is the smallest power of 2 greater than or equal to p. To understand the cost estimate (3.3), we remark that high-order lifting requires $O(\log \bar{p})$ lifting steps, each step requiring the multiplication of $\lceil m\bar{p}/n \rceil$ pairs of square matrices of dimension $\bar{n}$, each with degree bounded by $\deg X$.

We can simplify the asymptotic upper bound (3.3) as follows.

- We have $p = 1 + \lfloor d/\deg X \rfloor \leq 1 + d/\lceil md/n \rceil \leq 1 + n/m$. Using $m \leq n$ gives $p \leq 2n/m$. Using $\bar{p} < 2p$ gives $\bar{p} < 4n/m$. Thus we may substitute $\log \bar{p} \to \log(n/m)$.
- Using $\bar{p} < 4n/m$ and $n \leq \bar{n}$ we have $\lceil m\bar{p}/\bar{n} \rceil \leq 4$. Thus we may substitute $\lceil m\bar{p}/\bar{n} \rceil \to 1$.
- As before, use $\mathsf{PM}(\bar{n}, \deg X) \in O(\mathsf{PM}(n, nd/m))$.

The above simplifications yield a cost of

$$O(\log(n/m)\, \mathsf{PM}(n, md/n)) \tag{3.4}$$

operations in K to compute the X-adic expansion of the solution of (3.2) up to precision $\bar{p}$. The last step is to convert the X-adic expansion of the first m rows of $x^d F^{-1} \in \mathsf{K}[x]^{n\times n}$ to x-adic form. This is accomplished in time $O(\log(n/m)\, nm\, \mathsf{M}(d))$ operations in K using fast radix conversion [33, Theorem 9.15]. □

The next lemma establishes the second part of Theorem 3.5.

Lemma 3.9. *Let $F \in \mathsf{K}[x]^{n\times n}$ be as in Lemma 3.6. If $md < n$ then the first m rows of F^{-1} can be computed in $O(\log(d)(md)^\omega)$ field operations in K.*

Proof. If $md < n$ then F has at least $n - md$ columns of degree 0 by Lemma 3.6.2; since F is in Popov form, such columns have a 1 on the matrix's diagonal and are 0 on the remaining entries. The following describes how to essentially ignore $n - md$ of these columns.

Let P be a permutation matrix such that

$$\hat{F} := PFP^\top = \left[\begin{array}{c|c} F_1 & \\ \hline F_2 & I_{(n-md)\times(n-md)} \end{array}\right].$$

Let $\boldsymbol{v}$ be the first m rows of $x^d I_{n\times n}$. Our goal is to compute $\boldsymbol{v}F^{-1}$. Since

$$\hat{F}^{-1} = \left[\begin{array}{c|c} I_{md\times md} & \\ \hline -F_2 & I_{(n-md)\times(n-md)} \end{array}\right] \left[\begin{array}{c|c} F_1^{-1} & \\ \hline & I_{(n-md)\times(n-md)} \end{array}\right],$$

and $F^{-1} = P^\top \hat{F}^{-1} P$, we can factor the computation of $\boldsymbol{v}F^{-1}$ as follows:

$$\boldsymbol{v}F^{-1} = \left(\boldsymbol{v}P^\top \left[\begin{array}{c|c} I_{md\times md} & \\ \hline -F_2 & I_{(n-md)\times(n-md)} \end{array}\right]\right) \left[\begin{array}{c|c} F_1^{-1} & \\ \hline & I_{(n-md)\times(n-md)} \end{array}\right] P.$$

Let $\boldsymbol{v}_1 \in \mathsf{K}[x]^{m\times md}$ and $\boldsymbol{v}_2 \in \mathsf{K}[x]^{m\times(n-md)}$ be such that

$$\left[\begin{array}{c|c} \boldsymbol{v}_1 & \boldsymbol{v}_2 \end{array}\right] = \boldsymbol{v}P^\top \left[\begin{array}{c|c} I_{md\times md} & \\ \hline -F_2 & I_{(n-md)\times(n-md)} \end{array}\right].$$

Note that due to the structure of $\boldsymbol{v}$ and $P^\top$, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ can be constructed without any operations from K. We have thus reduced the computation of $\boldsymbol{v}F^{-1}$ to the following: $\boldsymbol{v}F^{-1} = \left[\begin{array}{c|c} \boldsymbol{v}_1 F_1^{-1} & \boldsymbol{v}_2 \end{array}\right] P$. As in the proof of Lemma 3.8, we will now use high-order lifting combined with partial linearisation to compute $\boldsymbol{v}_1 F_1^{-1}$.

The partial linearisation of $F_1 \in \mathsf{K}[x]^{md\times md}$ will have dimension $< 2md$ and degree 1. The lifting modulus used to solve the system is now $x-1$, and we need to lift up to precision $1+d$. Similar to before, the cost of the lifting is $O(\log(d)\,\mathsf{PM}(md,1))$, which is $O(\log(d)(md)^\omega)$. The radix conversion to convert the $(x-1)$-adic representation of $\boldsymbol{v}_1$ to x-adic representation has cost $O((\log d)m^2 d\,\mathsf{M}(d))$. Using the assumption $\mathsf{M}(d) \in O(d^{\omega-1})$ we see that $m^2 d\,\mathsf{M}(d) \in O((md)^\omega)$. □

3.3 The dual of a simultaneous Hermite Padé problem

Theorem 3.10. *Let $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$ be a instance of Problem 1.1 of size $t \times n$. Let A and B be as follows.*

$$A = \left[\begin{array}{c|c|c} I_{t\times t} & \boldsymbol{S} & \\ \hline & -\mathrm{diag}(\boldsymbol{g}) & I_{n\times n} \end{array}\right] \in \mathsf{K}[x]^{(t+n)\times(t+2n)} \qquad B = \left[\begin{array}{c} -\boldsymbol{S} \\ \hline I_{n\times n} \\ \hline \mathrm{diag}(\boldsymbol{g}) \end{array}\right] \in \mathsf{K}[x]^{(t+2n)\times n}$$

If G is a right $\boldsymbol{s}$-minimal approximant basis for A of order d with shift $\boldsymbol{s} \in \mathbb{Z}_{\geq 0}^{2n+t}$, then $x^d G^{-1}$ is a polynomial matrix and is a left $(-\boldsymbol{s})$-minimal approximant basis for B of order d. Moreover, if $\boldsymbol{\eta} = \mathrm{coldeg}_{\boldsymbol{s}} G$, then $\mathrm{rowdeg}_{(-\boldsymbol{s})}(x^d G^{-1}) = (d-\eta_1, \ldots, d-\eta_{2n+t})$.

Proof. Consider the following super-matrices of A respectively B:

$$\hat{A} = \left[\begin{array}{c|c|c} I_{t\times t} & \boldsymbol{S} & \\ \hline & x^d I_{n\times n} & \\ \hline & -\mathrm{diag}(\boldsymbol{g}) & I_{n\times n} \end{array}\right] \qquad \hat{B} = \left[\begin{array}{c|c|c} x^d I_{t\times t} & -\boldsymbol{S} & \\ \hline & I_{n\times n} & \\ \hline & \mathrm{diag}(\boldsymbol{g}) & x^d I_{n\times n} \end{array}\right]$$

Clearly G is also a right $\boldsymbol{s}$-minimal approximant basis for $\hat{A}$ of order d. Likewise, $\hat{B}$ and B have the same left minimal approximant basis for given order and shift. But direct computation shows that $\hat{A}\hat{B} = x^d I_{(2n+t)\times(2n+t)}$, and so the first part of Proposition 3.4 says that $\hat{A}$ is a left approximant basis for $\hat{B}$ of order d, and $\hat{B}$ is a right approximant basis of $\hat{A}$ of order d. The rest of the theorem now follows from Proposition 3.4. □

The idea is now to use Theorem 3.10: compute a left minimal approximant basis for B by computing a right minimal approximant basis G for A, and then use Theorem 3.5 to efficiently compute the first t columns of the $x^d G^{-1}$. But we first need to efficiently compute a right minimal approximant basis for A.

We accomplish this using Lemma 2.2: partition A into A_1, A_2 as follows:

$$A = \left[\begin{array}{c} A_2 \\ \hline A_1 \end{array}\right] = \left[\begin{array}{c|c|c} I_{t\times t} & \boldsymbol{S} & \\ \hline & -\mathrm{diag}(\boldsymbol{g}) & I_{n\times n} \end{array}\right] \in \mathsf{K}[x]^{(t+n)\times(t+2n)} .$$

We first compute a right minimal approximant basis G_1 for A_1. Lemma 3.11 describes how this can be done efficiently and that G_1 has a very simple shape. This allows us to efficiently compute a right minimal approximant basis for $A_2 G_1$.

Lemma 3.11. *Let $\boldsymbol{g} \in \mathsf{K}[x]^n$ be a vector of polynomials and let $\boldsymbol{s} \in \mathbb{Z}^{2n+t}$ be a shift. Let $P \in \mathsf{K}^{2n\times 2n}$ be the permutation matrix such that*

$$\left[\begin{array}{c|c} -\mathrm{diag}(\boldsymbol{g}) & I_{n\times n} \end{array}\right] P = \begin{bmatrix} -g_1 & 1 & & & & \\ & & -g_2 & 1 & & \\ & & & & \ddots & \\ & & & & & -g_n \quad 1 \end{bmatrix} \tag{3.5}$$

For $i = 1, \ldots, n$ let $H_i \in \mathsf{K}[x]^{2\times 2}$ be a right $\boldsymbol{s}_i$-minimal approximant basis of $\begin{bmatrix} -g_i & 1 \end{bmatrix} \in \mathsf{K}[x]^{1\times 2}$, where $\boldsymbol{s}_i = (s_{t+i}, s_{t+n+i})$, and let $\boldsymbol{h}_i := \mathrm{coldeg}_{\boldsymbol{s}_i} H_i$. Then a right minimal approximant basis of the matrix $[0_{n\times n} \mid \mathrm{diag}(\boldsymbol{g}) \mid I_{n\times n}]$ is given by G_1 where

$$G_1 = \begin{bmatrix} I_{t\times t} & \\ & P \end{bmatrix} \begin{bmatrix} I_t & & & \\ & H_1 & & \\ & & \ddots & \\ & & & H_n \end{bmatrix} , \tag{3.6}$$

with

$$\mathrm{coldeg}_{\boldsymbol{s}} G_1 = \big((s_1, \ldots, s_t) \mid \boldsymbol{h}_1 \mid \ldots \mid \boldsymbol{h}_n\big) .$$

Algorithm 2 DualitySimPade

Input: $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$, an instance of Problem 1.3 of size $t \times n$.
Output: $(\boldsymbol{\lambda}, \boldsymbol{\delta})$, solution specification.

$T \leftarrow \max_i T_i$
$d \leftarrow T + \max_i \deg g_i - 1$
$\big(H_i, \boldsymbol{h}_i\big) \leftarrow \mathsf{PopovMinBasis}_{\mathsf{Right}}\big(d, \big[-g_i\ 1\big], (N_i, T-1)\big)$ for $i = 1, \ldots, n$
$(G_1, \boldsymbol{h}) \leftarrow$ as in (3.6) with P as in (3.5)
$A_2 \leftarrow [I_t \ \ \boldsymbol{S} \ \ \mathbf{0}_{t\times n}] \in \mathsf{K}[x]^{t\times(2n+t)}$
$(G_2, \boldsymbol{\eta}) \leftarrow \mathsf{PopovMinBasis}_{\mathsf{Right}}(d, A_2 G_1, \boldsymbol{h})$
$\hat{\boldsymbol{\lambda}} \leftarrow$ first t columns of $x^d G_2^{-1}$
$\hat{\boldsymbol{\delta}} \leftarrow (d - \eta_1, \ldots, d - \eta_{n+1})$
$I \leftarrow \{i \mid \hat{\delta}_i < 0\}$, and $k \leftarrow |I|$
$(\boldsymbol{\lambda},\ \boldsymbol{\delta}) \leftarrow \big(\hat{\boldsymbol{\lambda}}_{i\in I},\ (\hat{\boldsymbol{\delta}}_i)_{i\in I}\big) \in \mathsf{K}[x]^{k\times t} \times \mathbb{Z}^k$
return $(\boldsymbol{\lambda}, \boldsymbol{\delta})$

Proof. Note first that permuting columns by P only has the effect on right $\boldsymbol{s}$-minimal approximant basis of permuting their rows by P^{-1}. The lemma then follows from repeated application of the easy observation, that if M_1 resp. M_2 is a right minimal approximant basis of C_1 resp. C_2, then

$$M = \left[\begin{array}{c|c} M_1 & \\ \hline & M_2 \end{array}\right]$$

is a right minimal approximant basis of

$$C = \left[\begin{array}{c|c} C_1 & \\ \hline & C_2 \end{array}\right]$$

. □

All the above is collected into Algorithm 2.

Theorem 3.12. *Algorithm 2 is correct. Let $d = \max_i T_i + \max_j \deg g_j - 1$. If $t < n$, then in terms of operations from K, the cost of the algorithm is*

1. $O(\mathsf{PM}(n, td/n)(\log(td/n)^2 + \log(n/t)) + n^{\omega-1} td \log(n) + nt\,\mathsf{M}(d)\log(n/t) + n\mathsf{M}(d)\log(d)^2)$.
2. $O(n(td)^{\omega-1} + (td)^\omega \log(d))$ *if* $td \in O(n)$.

Proof. Let $\boldsymbol{s} = (\boldsymbol{N} \mid T-1, \ldots, T-1)$ and d as in the algorithm. By combining Theorem 2.7 and Theorem 3.10, then if $(F, \boldsymbol{f}) = \mathsf{PopovMinBasis}_{\mathsf{Right}}(d, A, \boldsymbol{s})$ then the submatrix of $x^d F^{-1}$ comprised of those rows with negative $(-\boldsymbol{s})$-degree form a solution specification to the simultaneous Hermite Padé approximation. By combining Lemma 2.4 and Lemma 3.11, then $G_1 G_2$ is a right $\boldsymbol{s}$-minimal approximant basis of A with $\mathrm{coldeg}_{\boldsymbol{s}}(G_1 G_2) = \boldsymbol{\eta}$. A solution

specification is then given as the negative part of the first t columns of $x^d(G_1G_2)^{-1}$. Note that the first t columns of G_1 is $[I_t \mid \mathbf{0}]^\top$, so the first t columns of $x^d(G_1G_2)^{-1}$ are just the first t columns of $x^d G_2^{-1}$, as assigned to $\hat{\boldsymbol{\lambda}}$ in Line 7. By Theorem 3.10 then $\hat{\boldsymbol{\delta}}$ is the $(-\boldsymbol{s})$-row degrees of $x^d(G_1G_2)^{-1}$. The returned tuple $(\boldsymbol{\lambda}, \boldsymbol{\delta})$ is therefore a solution specification.

We estimate the complexity for the computationally expensive lines. Since $t < n$ we may use $n+t \in O(n)$. Line 3 costs n times $O(\mathsf{M}(d)\log(d)^2)$ by Theorem 2.6. Line 6 involves the product A_1G_1 and the call to $\mathsf{PopovMinBasis}_{\mathsf{Right}}$. The former costs $O(nt\,\mathsf{M}(d))$ due to the shape of G_1 according to (3.6), and the latter costs $O(\mathsf{PM}(n, td/n)\log(td/n)^2 + n^{\omega-1}td\log(n))$ by Theorem 2.6. Lastly, Line 7 costs $O(\log(n/t)(\mathsf{PM}(n, td/n) + nt\,\mathsf{M}(d)))$ by Theorem 3.5 since G_2 is the output of $\mathsf{PopovMinBasis}_{\mathsf{Right}}$.

Similarly, the second complexity estimate for the case $td \in O(n)$ follows from the second parts of Theorem 2.6 and Theorem 3.5. □

Example 3.13. We apply Algorithm 2 to the problem of Example 1.2 with shifts $\boldsymbol{N} = (5, 3 \mid 2, 3, 4, 4)$. We have

$$A \trianglelefteq \begin{bmatrix} 0 & & 4 & 4 & 4 & 4 & & & & \\ & 0 & 4 & 4 & 4 & 4 & & & & \\ & & 5 & & & & 0 & & & \\ & & & 5 & & & & 0 & & \\ & & & & 5 & & & & 0 & \\ & & & & & 5 & & & & 0 \end{bmatrix} \qquad B \trianglelefteq \begin{bmatrix} 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \\ 0 & & & \\ & 0 & & \\ & & 0 & \\ & & & 0 \\ 5 & & & \\ & 5 & & \\ & & 5 & \\ & & & 5 \end{bmatrix}$$

By Theorem 2.7 we are interested in an $(-\boldsymbol{s})$-minimal approximant basis for B of order $d = 5 + T - 1 = 9$, where $\boldsymbol{s} = (\boldsymbol{N} \mid T-1, \dots, T-1)$ and $T = \max T_i = 5$. By Theorem 3.10 such a basis is given as $x^d F^{-1}$, if F is a right $\boldsymbol{s}$-minimal approximant basis for A of order 9. We compute such an F as the product G_1G_2, where $(G_1, \boldsymbol{h}) = \mathsf{PopovMinBasis}_{\mathsf{Right}}(d, A_2, \boldsymbol{s})$, and $(G_2, \boldsymbol{\eta}) = \mathsf{PopovMinBasis}_{\mathsf{Right}}(d, A_1G_1, \boldsymbol{h})$. Such G_1 and G_2 are given by

$$G_1 \trianglelefteq \begin{pmatrix} 0 & & & & & & & & & \\ & 0 & & & & & & & & \\ & & 5 & & & & 4 & & & \\ & & & 5 & & & & 4 & & \\ & & & & 5 & & & & 4 & \\ & & & & & 5 & & & & 4 \\ & & 0 & & & & 4 & & & \\ & & & 0 & & & & 4 & & \\ & & & & 0 & & & & 4 & \\ & & & & & 0 & & & & 4 \end{pmatrix}, G_2 \trianglelefteq \begin{bmatrix} 5 & 4 & 3 & 4 & 4 & 3 & 4 & 4 & 3 & 4 \\ 4 & 6 & 5 & 0 & 5 & 5 & 4 & 4 & 3 & 4 \\ 0 & 1 & 2 & 0 & 1 & 0 & 0 & & 1 & \\ 0 & 0 & & 1 & 0 & & & 0 & 0 & \\ & & & & 0 & & & & & \\ 0 & & & & & 1 & & & & 0 \\ 0 & & 0 & & 0 & 0 & 1 & & 0 & \\ & 0 & 0 & & 0 & & & 1 & 0 & \\ & & & & & & & & 0 & \\ 0 & & 0 & & 0 & & & & & 1 \end{bmatrix}.$$

The $(-\boldsymbol{s})$-row-degrees of $x^d F^{-1}$ are $(d - \eta_1, \dots, d - \eta_{2n+t})$ where

$$\boldsymbol{\eta} = \operatorname{coldeg}_{\boldsymbol{s}} G_1G_2 = \operatorname{coldeg}_{\boldsymbol{h}} G_2 = (10,\ 9,\ 9,\ 9,\ 9,\ 10,\ 9,\ 9,\ 8,\ 9)$$

Thus, the first 2 columns of the submatrix of $x^d F^{-1}$ of rows 1 and 6 correspond to a solution specification:

$$\left[\begin{array}{c} (x^d F^{-1})_1 \\ \hline (x^d F^{-1})_6 \end{array}\right] \trianglelefteq \left[\begin{array}{cccccccccc} 4 & 2 & 0 & 2 & 3 & 1 & 3 & 3 & 3 & 3 \\ 3 & 1 & 0 & 1 & 1 & 3 & 2 & 2 & 2 & 2 \end{array}\right]$$

$x^d F^{-1}$ and $x^d G_2^{-1}$ agree on the first two columns, and so we use Theorem 3.5 to compute these efficiently.

4 Algorithm 2: Divide and Conquer

We now present our second algorithm for solving a $t \times n$ simultaneous Hermite Padé approximation. To describe the principle, consider a $t \times 2$ problem: first we compute solution basis to the 2 single $t \times 1$ Hermite Padé approximations, one for each column of the input $\boldsymbol{S} \in \mathsf{K}[x]^{t \times 2}$. This yields solution specifications $(\boldsymbol{\lambda}_1, \boldsymbol{\delta}_1) \in \mathsf{K}[x]^{k_1 \times t} \times \mathbb{Z}_{\geq 0}^{k_1}$ and $(\boldsymbol{\lambda}_2, \boldsymbol{\delta}_2) \in \mathsf{K}[x]^{k_2 \times t} \times \mathbb{Z}_{\geq 0}^{k_2}$. We then need to *intersect* these solutions, in the following sense: any $(\boldsymbol{\lambda} \mid \boldsymbol{\phi}) \in \mathsf{K}[x]^{1 \times (t+2)}$ which is a solution to both single Hermite Padé approximations must satisfy that $\boldsymbol{\lambda}$ is in the row space of both $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$. Further, since the completions of either $\boldsymbol{\lambda}_i$ is a $(-\boldsymbol{N})$-row reduced matrix, the Predictable Degree property allows us to compute the $(-\boldsymbol{N})$-degree of $(\boldsymbol{\lambda} \mid \boldsymbol{\phi})$ by inspecting only the degrees of the linear combinations used to form $\boldsymbol{\lambda}$ from $\boldsymbol{\lambda}_1$ resp. $\boldsymbol{\lambda}_2$.

4.1 t-intersections of row spaces

Given two matrices $F_1, F_2 \in \mathsf{K}[x]^{* \times m}$, it is natural to consider computing a basis for the intersection of their row spaces. We could do that by computing a left kernel of the following matrix:

$$R = \left[\begin{array}{c|c} I_{m \times m} & I_{m \times m} \\ \hline F_1 & \\ & F_2 \end{array}\right] .$$

Note that any vector $(\boldsymbol{v} \mid \boldsymbol{b}_1 \mid \boldsymbol{b}_2)$ in the left kernel of R must satisfy $\boldsymbol{v} = \boldsymbol{b}_1 F_1$ and $\boldsymbol{v} = \boldsymbol{b}_2 F_2$, hence $\boldsymbol{v}$ is in the row space of both F_1 and F_2. If we needed only small vectors in the intersection, say $\deg \boldsymbol{v} < d$, and if the row space of F_1 and F_2 were row reduced and of degree less than d, then it would be cheaper to compute a left minimal approximant basis of R of order $2d$: firstly, all kernel vectors would of course be minimal approximants; and secondly, if $\boldsymbol{m}$ is a minimal approximant of R of order $2d$ with $\deg m < d$, then $\deg(vR) < 2d$ so the congruence lifts to an equality. Computing the negative part of the minimal approximant of R in row reduced form would even yield a row reduced basis of $\mathrm{Row}(F_1) \cap \mathrm{Row}(F_2)$ by its first m columns.

Consider now that F_1 is the first part of a larger matrix $A_1 = [F_1|H_1]$ and similarly $A_2 = [F_2|H_2]$. It is still natural to consider $\mathrm{Row}(F_1) \cap \mathrm{Row}(F_2)$, but now, for a vector $\boldsymbol{v} = \boldsymbol{b}_1 F_1 = \boldsymbol{b}_2 F_2$, it could be important to compute also $\boldsymbol{b}_1 H_1$ and $\boldsymbol{b}_2 H_2$. Alternatively, we might need to just compute a reduced basis of the intersection of F_1 and F_2, but still track degrees of the parts corresponding to H_1 and H_2.

Definition 4.1. Let $A_1 = [\boldsymbol{\lambda}_1 \mid H_1] \in \mathsf{K}[x]^{k_1 \times (t+n_1)}$ and $A_2 = [\boldsymbol{\lambda}_2 \mid H_2] \in \mathsf{K}[x]^{k_2 \times (t+n_2)}$. The *t-intersection* of A_1 and A_2 is the $\mathsf{K}[x]$-module:

$$\mathcal{I}_t(A_1, A_2) = \left\{ (\boldsymbol{\lambda} \mid \boldsymbol{a}_1 \mid \boldsymbol{a}_2) \;\middle|\; (\boldsymbol{\lambda} \mid \boldsymbol{a}_i) \in \mathrm{Row}(A_i) \text{ for } i = 1, 2 \right\} .$$

Consider shifts $\boldsymbol{h}_1 = (\boldsymbol{v} \mid \boldsymbol{s}_1) \in \mathbb{Z}^{t+n_1}$ and $\boldsymbol{h}_2 = (\boldsymbol{v} \mid \boldsymbol{s}_2) \in \mathbb{Z}^{t+n_1}$ sharing the first t components. If A_i is $\boldsymbol{h}_i$-row reduced for $i = 1, 2$, then an *$\boldsymbol{h}$-shifted t-intersection basis* of A_1 and A_2 is a matrix $P \in \mathsf{K}[x]^{k \times (t+n_1+n_2)}$ which is an $\boldsymbol{h}$-row reduced basis of $\mathcal{I}_t(A_1, A_2)$, where $\boldsymbol{h} = (\boldsymbol{v} \mid \boldsymbol{s}_1 \mid \boldsymbol{s}_2)$.

Theorem 4.2. *Consider shifts $\boldsymbol{h}_1 = (\boldsymbol{v} \mid \boldsymbol{s}_1) \in \mathbb{Z}^{t+n_1}$ and $\boldsymbol{h}_2 = (\boldsymbol{v} \mid \boldsymbol{s}_2) \in \mathbb{Z}^{t+n_1}$ sharing the first t components, and let $A_1 = [\boldsymbol{\lambda}_1 \mid H_1] \in \mathsf{K}[x]^{k_1 \times (t+n_1)}$ and $A_2 = [\boldsymbol{\lambda}_2 \mid H_2] \in \mathsf{K}[x]^{k_2 \times (t+n_2)}$ be $\boldsymbol{h}_1$- resp. $\boldsymbol{h}_2$- row reduced.*

Let $\boldsymbol{r} = \left(\boldsymbol{v} \mid \mathrm{rowdeg}_{\boldsymbol{h}_1}(H_1) \mid \mathrm{rowdeg}_{\boldsymbol{h}_2}(H_2)\right)$ and let M be an $\boldsymbol{r}$-row reduced kernel basis of R, where

$$R = \left[\begin{array}{c|c} I_{t\times t} & I_{t\times t} \\ \hline -\boldsymbol{\lambda}_1 & \\ & -\boldsymbol{\lambda}_2 \end{array}\right] .$$

Then MC is an $\boldsymbol{h}$-shifted t-intersection basis for A_1 and A_2, where

$$C = \begin{bmatrix} I_{t\times t} & & \\ & H_1 & \\ & & H_2 \end{bmatrix}$$

and $\mathrm{rowdeg}_{\boldsymbol{h}}(MC) = \mathrm{rowdeg}_{\boldsymbol{r}}(M)$*, where $\boldsymbol{h} = (\boldsymbol{v} \mid \boldsymbol{s}_1 \mid \boldsymbol{s}_2)$. In particular, the first t columns of M is the first t columns of an $\boldsymbol{h}$-shifted t-intersection basis.*

Proof. First note that due to the shape of R, if M' is the first t columns of M, then $\mathrm{Row}(M')$ is the set of vectors that lie in both $\mathrm{Row}(\boldsymbol{\lambda}_1)$ and $\mathrm{Row}(\boldsymbol{\lambda}_2)$. Thus the rows of MC really span the t-intersection of A_1 and A_2. To show that MC is $\boldsymbol{h}$-row reduced with $\mathrm{rowdeg}_{\boldsymbol{h}}(MC) = \mathrm{rowdeg}_{\boldsymbol{r}}(M)$, consider the following amalgamation of R and C:

$$F = \left[\begin{array}{cccc} I_{t\times t} & I_{t\times t} & & I_{t\times t} \\ \hline & -\boldsymbol{\lambda}_1 & -H_1 & \\ & & & -\boldsymbol{\lambda}_2 \quad -H_2 \end{array}\right] .$$

Since $[\boldsymbol{\lambda}_1 \mid H_1]$ and $[\boldsymbol{\lambda}_2 \mid H_2]$ are $\boldsymbol{h}_1$ resp. $\boldsymbol{h}_2$ row reduced, then F has full row rank and is $\boldsymbol{h}' = (\boldsymbol{v} \mid \boldsymbol{v} \mid \boldsymbol{s}_1 \mid \boldsymbol{v} \mid \boldsymbol{s}_2)$ row reduced. Note that M is $\boldsymbol{r}$-row reduced and $\boldsymbol{r} = \text{rowdeg}_{\boldsymbol{h}'}(F)$. Thus by Lemma 2.2 then MF is $\boldsymbol{h}'$-row reduced with $\text{rowdeg}_{\boldsymbol{h}'}(MF) = \text{rowdeg}_{\boldsymbol{r}}(M)$. But since M is a kernel basis of R, then MF is, up to negation of some columns, the same as MC with two blocks of t zero-columns inserted. Thus MC is $\boldsymbol{h}$-row reduced with $\text{rowdeg}_{\boldsymbol{h}}(MC) = \text{rowdeg}_{\boldsymbol{r}}(M)$. □

Since R of Theorem 4.2 has rank at least t, then this shows that a t-intersection of two matrices with row-dimension k_1 resp. k_2 has dimension up to $k_1 + k_2$.

In general, the kernel of R could have entries as large as $(k_1+k_2+t)\deg R$, so computing the full kernel could be expensive. In our application of solving simultaneous Hermite Padé approximations, however, we will only be needing the negative part of a shifted t-intersection basis, which means we need only compute the low-degree rows of M: but these will be contained in a shifted minimal approximant basis of R, as we will see. To do this we will also use the following lemma:

Lemma 4.3. *Consider shifts $\boldsymbol{h}_1 = (\boldsymbol{v} \mid \boldsymbol{s}_1) \in \mathbb{Z}^{t+n_1}$ and $\boldsymbol{h}_2 = (\boldsymbol{v} \mid \boldsymbol{s}_2) \in \mathbb{Z}^{t+n_1}$ sharing the first t components, and let $A_1 \in \mathsf{K}[x]^{k_1\times(t+n_1)}$ and $A_2 \in \mathsf{K}[x]^{k_2\times(t+n_2)}$ be $\boldsymbol{h}_1$- resp. $\boldsymbol{h}_2$- row reduced. Let B_i be the $\boldsymbol{h}_i$-shifted negative part of A_i for $i = 1, 2$. Then the negative part of the $\boldsymbol{h}$-shifted t-intersection of B_1 and B_2 equals the negative part of the $\boldsymbol{h}$-shifted t-intersection of A_1 and A_2.*

Proof. Assume oppositely that $\mathcal{I}_t(A_1, A_2) \setminus \mathcal{I}_t(B_1, B_2)$ contains a vector $\boldsymbol{v}$ with $\deg_{\boldsymbol{h}}(v) < 0$. Write $\boldsymbol{v} = (\boldsymbol{\lambda} \mid \boldsymbol{a}_1 \mid \boldsymbol{a}_2)$. Then either $(\boldsymbol{\lambda} \mid a_1) \notin \text{Row}(B_1)$ or $(\boldsymbol{\lambda} \mid a_2) \notin \text{Row}(B_2)$; assume wlog. the former. There must be a $\boldsymbol{q}$ such that $(\boldsymbol{\lambda}_v \mid \boldsymbol{a}_1) = \boldsymbol{q}A_1$ and further that $\boldsymbol{q}$ is non-zero on an index i corresponding to a row of A_1 which is not in B_1. But this row has non-negative $\boldsymbol{h}_1$-shifted degree, and so by the Predictable Degree property, so will $(\boldsymbol{\lambda} \mid a_1)$, contradicting that $\boldsymbol{v}$ has negative $\boldsymbol{h}$-shifted degree. □

4.2 Building up simultaneous Hermite Padé solutions

Consider a size $(t \times 2n)$ simultaneous Hermite Padé instance $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$ with $\boldsymbol{S} = (\boldsymbol{S}_1 \mid \boldsymbol{S}_2)$, $\boldsymbol{g} = (\boldsymbol{g}_1 \mid \boldsymbol{g}_2)$ and $\boldsymbol{N} = (\boldsymbol{T} \mid \boldsymbol{N}_1 \mid \boldsymbol{N}_2)$. By (2.1) of Section 2.4.1 then if $P = \mathsf{K}[x]^{(t+2n)\times(t+2n)}$ is an $(-\boldsymbol{N})$-row reduced basis of A, where

$$A = \left[\begin{array}{c|c} I_{t\times t} & \boldsymbol{S} \\ \hline & \text{diag}(\boldsymbol{g}) \end{array}\right] = \left[\begin{array}{c|cc} I_{t\times t} & \boldsymbol{S}_1 & \boldsymbol{S}_2 \\ \hline & \text{diag}(\boldsymbol{g}_1) & \\ & & \text{diag}(\boldsymbol{g}_2) \end{array}\right]$$

then the sub-matrix of P comprised of the rows with negative $(-\boldsymbol{N})$-degree form a solution basis. But the second form of A above demonstrates that P is

exactly a t-intersection basis of the two matrices $A_i = \left[\begin{array}{c|c} I_{t\times t} & \boldsymbol{S}_i \\ \hline & \operatorname{diag}(\boldsymbol{g}_i) \end{array}\right]$ for $i = 1, 2$, with shifts $-\boldsymbol{N}_1$ respectively $-\boldsymbol{N}_2$: for if $(\boldsymbol{\lambda}, \boldsymbol{a}_i) \in \operatorname{Row}(A_i), i = 1, 2$ then there are $\boldsymbol{q}_i \in \mathsf{K}[x]^{1\times n_i}$ such that $(\boldsymbol{\lambda} \mid \boldsymbol{q}_i)A_i = (\boldsymbol{\lambda} \mid \boldsymbol{a}_i)$ and hence $(\boldsymbol{\lambda} \mid \boldsymbol{q}_1 \mid \boldsymbol{q}_2)A = (\boldsymbol{\lambda} \mid \boldsymbol{a}_1 \mid \boldsymbol{a}_2)$, and vice versa. The intersections are then structured recursively in a Divide & Conquer tree.

Our recursive algorithm will return only the negative part of a reduced basis of each A_i, and not the entire basis, but this suffice to compute the negative part of the t-intersection, as according to Lemma 4.3.

Example 4.4. Consider again Example 1.2 and the execution of Algorithm 3 on this input. We divide the problem into two 2×2 simultaneous Hermite Padé problems $\boldsymbol{S}_1 \in \mathsf{K}[x]^{2\times 2}$, $\boldsymbol{N}_1 = (5, 3 \mid 2, 3)$, and $\boldsymbol{S}_2 \in \mathsf{K}[x]^{2\times 2}$ and $\boldsymbol{N}_2 = (5, 3 \mid 4, 4)$. Note that the first $t = 2$ positions on $\boldsymbol{N}_1$ and $\boldsymbol{N}_2$ agree, since this is the degree bound on the sought λ for the combined problem. The sub-problems have the following solution specifications and their completions:

$$\boldsymbol{\lambda}_1 \trianglelefteq \begin{bmatrix} 2 & 1 \\ 3 & 1 \end{bmatrix} \qquad \boldsymbol{\delta}_1 = [-1, -2] \qquad A_1 \trianglelefteq \begin{bmatrix} 2 & 1 & 1 & \\ 3 & 1 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{\lambda}_2 \trianglelefteq \begin{bmatrix} 3 & 1 \\ 4 & \\ & 2 \\ 2 & 0 \end{bmatrix} \qquad \boldsymbol{\delta}_2 = [-2, -1, -1, -2] \qquad A_2 \trianglelefteq \begin{bmatrix} 3 & 1 & 2 & \\ 4 & & & \\ & 2 & 3 & 3 \\ 2 & 0 & 2 & 1 \end{bmatrix}$$

We construct R as in Line 12. Let $\boldsymbol{r} = (-5, -3, -1, -2, -2, -1, -1, -2)$. Below is R as well as an $\boldsymbol{r}$-minimal approximant basis for R of order $T = 5$ in Popov form:

$$R \trianglelefteq \begin{pmatrix} 0 & & 0 & \\ & 0 & & 0 \\ 2 & 1 & & \\ 3 & 1 & & \\ & & 3 & 1 \\ & & 4 & \\ & & & 2 \\ & & 2 & 0 \end{pmatrix} \qquad G \trianglelefteq \begin{pmatrix} 5 & & & & & & & \\ & 5 & & & & & & \\ 4 & 1 & 2 & 2 & 0 & 0 & & \\ & & & 4 & & & & \\ 4 & 2 & 0 & 1 & 1 & & & 0 \\ 4 & 1 & 1 & 1 & & 1 & & 1 \\ 4 & 2 & 0 & 1 & 0 & 0 & 0 & \\ 3 & 2 & 1 & & 0 & 0 & & 2 \end{pmatrix},$$

where $\operatorname{rowdeg}_{\boldsymbol{s}}(G) = (0, 2, 1, 2, -1, 0, -1, 0)$. Only rows 5 and 7 have negative $\boldsymbol{r}$-degree, and only these will show up in NegMinBasis. The first two elements of each of those rows along with the shifted degrees $(-1, -1)$ comprise the solution specification:

$$\left[\begin{array}{c} \boldsymbol{\lambda}'_1 \\ \hline \boldsymbol{\lambda}'_2 \end{array}\right] = \begin{bmatrix} x^4 + x^3 + x & x^2 + 1 \\ x^4 & x^2 + x + 1 \end{bmatrix}$$

Note that $\boldsymbol{\lambda}'_1 = \boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}'_2 = \boldsymbol{\lambda}_1 + \boldsymbol{\lambda}_2$, where $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2$ is as in Example 1.2.

Algorithm 3 RecursiveSHPade

Input: $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$, an instance of Problem 1.3 of size $t \times n$.
Output: $(\boldsymbol{\lambda}, \boldsymbol{\delta}) \in \mathsf{K}[x]^{k\times t} \times \mathbb{Z}^k_{<0}$, a solution specification to $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$.

1 **if** $n \leq t$ **then**
2 **return** DirectSHPade$(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$
3 **else**
4 $(T_1, \ldots, T_t, N_1, \ldots, N_n) \leftarrow \boldsymbol{N}$
5 $\boldsymbol{S}_1, \boldsymbol{S}_2 \leftarrow \boldsymbol{S}$ split into $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$ columns
6 $\boldsymbol{g}_1, \boldsymbol{g}_2 \leftarrow \boldsymbol{g}$ split into $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$ elements
7 $\boldsymbol{N}_1 \leftarrow (T_1, \ldots, T_t, N_1, \ldots, N_{\lceil n/2 \rceil})$
8 $\boldsymbol{N}_2 \leftarrow (T_1, \ldots, T_t, N_{\lceil n/2 \rceil + 1}, \ldots, N_n)$
9 $(\boldsymbol{\lambda}_1, \boldsymbol{\delta}_1) \leftarrow \mathsf{RecursiveSHPade}\big(\boldsymbol{S}_1, \boldsymbol{g}_1, \boldsymbol{N}_1\big)$
10 $(\boldsymbol{\lambda}_2, \boldsymbol{\delta}_2) \leftarrow \mathsf{RecursiveSHPade}\big(\boldsymbol{S}_2, \boldsymbol{g}_2, \boldsymbol{N}_2\big)$
11 $\boldsymbol{r} \leftarrow (-T_1, \ldots, -T_t \mid \boldsymbol{\delta}_1 \mid \boldsymbol{\delta}_2)$
12 $R \leftarrow \left[\begin{array}{c|c} I_t & I_t \\ \hline -\boldsymbol{\lambda}_1 & \\ \hline & -\boldsymbol{\lambda}_2 \end{array}\right]$
13 $(\left[\begin{array}{c|c} \boldsymbol{\lambda} & * \end{array}\right], \boldsymbol{\delta}) \leftarrow \mathsf{NegMinBasis}(\max_j T_j, R, \boldsymbol{r})$ where $\boldsymbol{\lambda} \in \mathsf{K}[x]^{*\times t}$
14 **return** $(\boldsymbol{\lambda}, \boldsymbol{\delta})$
15 **end if**

Theorem 4.5. *Algorithm 3 is correct. Let $d = \max_i T_i + \max_i \deg g_i$. In terms of field operations from K, and assuming $t < n$, it has complexity*

1. $O\big(\mathsf{PM}(n, td/n)\log(td/n)^2 + (n/t)\mathsf{PM}(t,d)\log(d)^2 + n^{\omega-1}td\log(n)\big)$.
2. $O\big((n/t)\mathsf{PM}(t,d)\log(d)^2 + n(td)^{\omega-1}\log(n)\big)$ *when* $td \in O(n)$.

Proof. Correctness is established by induction on n. The base case is correct by the correctness of DirectSHPade.

For the recursive case, let P'_i be the completion of $\boldsymbol{\lambda}_i$ for $i = 1, 2$, and note that $\mathrm{rowdeg}_{-\boldsymbol{N}_i}(P'_i) = \boldsymbol{\delta}_i$. Note that P'_i is the negative part of some $(-\boldsymbol{N}_i)$-row reduced matrix P_i which is row-equivalent to $A_i = \left[\begin{array}{c|c} I_{t\times t} & \boldsymbol{S}_i \\ \hline & \mathrm{diag}(\boldsymbol{g}_i) \end{array}\right]$, as according to Section 2.4. By the induction hypothesis and from the discussion at the beginning of the section, then if P is an $(-\boldsymbol{N})$-shifted t-intersection basis of P_1 and P_2, then P is a solution to the problem instance. By Lemma 4.3, we can get the $(-\boldsymbol{N})$-shifted negative part of P as a t-intersection of just the $(-\boldsymbol{N}_i)$-shifted negative part of P_1 resp. P_2, i.e. P'_1 and P'_2. For a solution specification, we need just the first t columns of such an intersection basis, and by Theorem 4.2, we get this as the first t columns of an $\boldsymbol{r}$-shifted left kernel of R.

Left is therefore only to prove that Line 13 actually computes the negative part of an $\boldsymbol{r}$-row reduced kernel basis of R, that is, we should prove that

each row in $\mathsf{NegMinBasis}(T, R, \boldsymbol{r})$ is in fact a kernel vector (since kernel vectors are clearly minimal approximants). So let $\boldsymbol{w} = (\boldsymbol{\lambda} \mid \boldsymbol{w}_1 \mid \boldsymbol{w}_2)$ be a minimal approximant of R of order T with $\deg_{\boldsymbol{r}} \boldsymbol{w} < 0$. Then $\boldsymbol{w}R = (\boldsymbol{\lambda} - \boldsymbol{w}_1\boldsymbol{\lambda}_1, \boldsymbol{\lambda} - \boldsymbol{w}_2\boldsymbol{\lambda}_2)$. Since $\deg_{\boldsymbol{r}} \boldsymbol{w} < 0$ then $\deg \boldsymbol{\lambda} < T$ and for $i = 1, 2$ then $\operatorname{coldeg} \boldsymbol{w}_i < -\boldsymbol{\delta}_i$. But also $\operatorname{rowdeg}_{(-T,\ldots,-T)} \boldsymbol{\lambda}_i \leq \boldsymbol{\delta}_i$ since $\boldsymbol{\lambda}_i$ are the solutions to the i'th sub-problem. We conclude $\deg(\boldsymbol{w}_i\boldsymbol{\lambda}_i) < T$ and thus $\deg(\boldsymbol{w}R) < T$. But since $\boldsymbol{w}R \equiv 0 \mod x^T$ we must have $\boldsymbol{w}R = 0$.

For complexity, we let $C(n)$ be the cost Algorithm 3 for given n. For the base case $n \leq t$ we use Corollary 2.8. For the recursive step $n > t$ we use Theorem 2.6 and recall that each of $(\boldsymbol{\lambda}_i, \boldsymbol{\delta}_i)$ have at most $t + \lceil n/2 \rceil$ entries since they are solution specifications to problems of size roughly $t \times \lceil n/2 \rceil$. This also means that the degree of R in Line 13 is at most $T := \max\{T_1, \ldots, T_t\}$. The call to $\mathsf{NegMinBasis}$ in Line 13 uses an order bounded by T, but for simplicity we will use the upper bound $d := T + \max \deg g_i$. Then we get the following recursion on $C(n)$.

$$C(n) = \begin{cases} 2C(n/2) + O\big(n(td)^{\omega-1} + (td)^{\omega}\log(d)\big) & \text{if } n \geq td \\ 2C(n/2) + O\big(\mathsf{PM}(n, td/n)\log(td/n)^2 + n^{\omega-1}td\log(n)\big) & \text{if } t < n < td \\ O\big(\mathsf{PM}(t, d)\log(d)^2 + t^{\omega}d\log(t)\big) & \text{if } n \leq t \end{cases},$$

The total cost at the $O(n/t)$ leaf nodes of the recursion tree corresponding to the base case $n \leq t$ is

$$O\Big((n/t)\mathsf{PM}(t, d)\log(d)^2 + nt^{\omega-1}d\log(t)\Big). \tag{4.1}$$

If $n < td$ then the case $n \geq td$ of the recurrence never occurs. By the Master Theorem, using our assumption $\omega > 2$ and $\mathsf{PM}(n, td/n) \in \Omega(n^{\omega-1}td)$, the total work done at the internal nodes of the recursion tree corresponding to the case $t < n < td$ will be dominated by the work done at the root node:

$$O(\mathsf{PM}(n, td/n)\log(td/n)^2 + n^{\omega-1}td\log(n)). \tag{4.2}$$

Summing (4.1) and (4.2) and noting that $nt^{\omega-1}d\log(t) \in O(n^{\omega-1}td\log(n))$ since $t < n$ shows that when $n < td$ then:

$$C(n) \in O\big(\mathsf{PM}(n, td/n)\log(td/n)^2 + (n/t)\mathsf{PM}(t, d)\log(d)^2 + n^{\omega-1}td\log(n)\big)\ . \tag{4.3}$$

If $n \geq td$ then let $k \in \Theta(\log(n/(td)))$ be the largest integer such that $n/2^k \geq td$. Then exactly the first k levels in the recursion tree correspond to the case $n \geq td$ of the recurrence. Summing the total work done at all $O(2^k)$ nodes in the first k levels of the recursion tree and using $2^k \in \Theta(n/(td))$ gives

$$O(\log(n/(td))n(td)^{\omega-1} + n(td)^{\omega-1}\log(d)).$$

Using the Master Theorem as before, the work done at internal nodes corresponding to the case $t < n < td$ of the recurrence (i.e., internal nodes

at all levels $> k$) will be dominated by the sum of the work done at the $2^{k+1} \in O(n/(td))$ nodes at the single level $k+1$:

$$O(n(td)^{\omega-1}\log(td)).$$

Summing the last two cost bounds, and using $n \geq d$, shows the total work done at the internal nodes of the recursion tree the case $n \geq td$ is bounded by $O(n(td)^{\omega-1}\log(n))$; summing this last bound with (4.1) and noting that $nt^{\omega-1}d\log(t) \in O(n(td)^{\omega-1}\log(td))$ shows that

$$C(n) \in O\big((n/t)\mathsf{PM}(t,d)\log(d)^2 + n(td)^{\omega-1}\log(n)\big) \tag{4.4}$$

when $td \leq n$.

Finally, if $td \leq n$ then (4.4) is "big-O" of (4.3) and thus (4.3) holds also when $td \leq n$. □

It is interesting to note that in Algorithm 3 we compute the first t columns of the negative part of a t-intersection basis of the completions of $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$, each of which could have row-dimensions up to roughly $t + n/2$; thus we would expect that the t-intersection has row-dimension up to $2t + n$. But this will not happen since we proved that the output of the algorithm is a solution specification to the input $t \times n$ simultaneous Hermite Padé problem, and one such can have at most $t + n$ entries.

Acknowledgements. The authors would like to thank George Labahn for valuable discussions, and for making us aware of the Hermite–Simultaneous Padé duality. We would also like to thank Vincent Neiger for making preprints of [19] available to us.

References

[1] G. Baker and P. Graves-Morris. *Padé approximants*, volume 59. Cambridge Univ. Press, 1996.

[2] M. V. Barel and A. Bultheel. A general module theoretic framework for vector M-Padé and matrix rational interpolation. *Numerical Algorithms*, 3(1):451–461, Dec. 1992.

[3] B. Beckermann. A reliable method for computing M-Padé approximants on arbitrary staircases. *J. Comp. App. Math.*, 40(1):19–42, June 1992.

[4] B. Beckermann and G. Labahn. A uniform approach for Hermite Padé and simultaneous Padé approximants and their matrix-type generalizations. *Numerical Algorithms*, 3(1):45–54, 1992.

[5] B. Beckermann and G. Labahn. A Uniform Approach for the Fast Computation of Matrix-Type Padé Approximants. *SIAM J. Matr. Anal. Appl.*, 15(3):804–823, July 1994.

[6] B. Beckermann and G. Labahn. Recursiveness in matrix rational interpolation problems. *J. Comp. App. Math.*, 77(1–2):5–34, Jan. 1997.

[7] B. Beckermann and G. Labahn. Fraction-Free Computation of Simultaneous Padé Approximants. In *Proc. of ISSAC*, pages 15–22, 2009.

[8] B. Beckermann, G. Labahn, and G. Villard. Normal forms for general polynomial matrices. *J. Symb. Comp.*, 41(6):708–737, 2006.

[9] E. R. Berlekamp. *Algebraic Coding Theory.* Aegean Park Press, 1968.

[10] A. Bostan, C.-P. Jeannerod, and E. Schost. Solving structured linear systems with large displacement rank. *Th. Comp. Sc.*, 407(1–3):155–181, Nov. 2008.

[11] D. G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica*, 28(7):693–701, 1991.

[12] D. Coppersmith and S. Winograd. Matrix Multiplication via Arithmetic Progressions. *J. Symb. Comp.*, 9(3):251–280, 1990.

[13] F. L. Gall. Powers of tensors and fast matrix multiplication. In *Proc. of ISSAC*, pages 296–303, 2014.

[14] P. Giorgi, C. Jeannerod, and G. Villard. On the Complexity of Polynomial Matrix Computations. In *Proc. of ISSAC*, pages 135–142, 2003.

[15] S. Gupta, S. Sarkar, A. Storjohann, and J. Valeriote. Triangular x-basis decompositions and derandomization of linear algebra algorithms over $\mathsf{K}[x]$. *J. Symb. Comp.*, 47(4):422–453, 2012.

[16] F. Gustavson and D. Yun. Fast algorithms for rational Hermite approximation and solution of Toeplitz systems. *IEEE Trans. Circ. Sys.*, 26(9):750–755, 1979.

[17] D. Harvey, J. V. D. Hoeven, and G. Lecerf. Faster polynomial multiplication over finite fields. *Journal of the ACM*, 63(6), Feb. 2017.

[18] C. Hermite. Sur la Formule d'Interpolation de Lagrange. *J. Reine und Angewandte Math.*, 84(1):70–79, 1878.

[19] C.-P. Jeannerod, V. Neiger, E. Schost, and G. Villard. Fast computation of minimal interpolation bases in Popov form for arbitrary shifts. Submitted to ISSAC'16.

[20] C.-P. Jeannerod, V. Neiger, E. Schost, and G. Villard. Fast Computation of Minimal Interpolation Bases in Popov Form for Arbitrary Shifts. In *International Symposium on Symbolic and Algebraic Computation*, ISSAC '16, pages 295–302, New York, NY, USA, 2016. ACM.

[21] C.-P. Jeannerod, V. Neiger, É.. Schost, and G. Villard. Computing minimal interpolation bases. *J. Symb. Comp.*, 83:272–314, 2017.

[22] C.-P. Jeannerod, V. Neiger, and G. Villard. Fast computation of approximant bases in canonical form. Jan. 2018. arXiv: 1801.04553.

[23] J. Justesen. On the complexity of decoding Reed-Solomon codes (Corresp.). *IEEE Trans. Inf. Theory*, 22(2):237–238, Mar. 1976.

[24] T. Kailath. *Linear Systems*. Prentice-Hall, 1980.

[25] K. Mahler. Perfect systems. *Compos. Math*, 19:95–168, 1968.

[26] V. Neiger. Fast computation of shifted Popov forms of polynomial matrices via systems of modular polynomial equations. In *International Symposium on Symbolic and Algebraic Computation*, July 2016.

[27] V. Neiger and T. X. Vu. Computing Canonical Bases of Modules of Univariate Relations. In *International Symposium on Symbolic and Algebraic Computation*, page 8, July 2017.

[28] H. Padé. *Sur la représentation approchée d'une fonction par des fractions rationnelles*. Number 740. Gauthier-Villars et fils, 1892.

[29] J. Rosenkilde né Nielsen and A. Storjohann. Algorithms for Simultaneous Padé Approximations. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC '16, pages 405–412, New York, NY, USA, 2016. ACM.

[30] W. A. Stein et al. SageMath Software. http://www.sagemath.org.

[31] A. Storjohann. High-order lifting and integrality certification. *J. Symb. Comp.*, 36(3):613–648, 2003.

[32] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa. Further Results on Goppa Codes and their Applications to Constructing Efficient Binary Codes. *IEEE Trans. Inf. Theory*, 22(5):518–526, 1976.

[33] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge Univ. Press, 3rd edition, 2012.

[34] A. Zeh, C. Gentner, and D. Augot. An Interpolation Procedure for List Decoding Reed-Solomon Codes Based on Generalized Key Equations. *IEEE Trans. Inf. Theory*, 57(9):5946–5959, 2011.

[35] W. Zhou and G. Labahn. Efficient algorithms for order basis computation. *J. Symb. Comp.*, 47(7):793–819, 2012.