Algorithms for simultaneous Hermite–Padé approximations

Rosenkilde, Johan; Storjohann, Arne

Published in:
Journal of Symbolic Computation

Link to article, DOI:
10.1016/j.jsc.2019.07.026

Publication date:
2020

Document Version
Peer reviewed version

Citation (APA):
Rosenkilde, J., & Storjohann, A. (2020). Algorithms for simultaneous Hermite–Padé approximations. *Journal of Symbolic Computation*, *102*, 279-303. https://doi.org/10.1016/j.jsc.2019.07.026

Algorithms for Simultaneous Hermite–Padé Approximations

Johan Rosenkilde

Technical University of Denmark, Denmark

Arne Storjohann

University of Waterloo, Canada

Abstract

We describe how to compute simultaneous Hermite–Padé approximations, 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, where n is the number of simultaneous approximations using $t < n$ polynomials, and where $O(n^{\omega})$ is the cost of multiplying $n \times n$ matrices over K. We develop two algorithms using different approaches. Both algorithms return a reduced sub-basis that generates the complete set of solutions to the input approximation problem that satisfy the given degree constraints. Previously, the cost $O^{\sim}(n^{\omega-1}td)$ has only been reached with randomized algorithms finding a single solution for the case $t < n$. Our results are made possible by recent breakthroughs in fast computations of minimal approximant bases and Hermite–Padé approximations for the case $t \geq n$.

Keywords: Padé approximation; approximant bases; structured linear systems

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 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 $\lambda_1 S_1 + \ldots + \lambda_t S_t \equiv \phi \mod x^d$. In vector form:

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

Email addresses: jsrn@jsrn.dk (Johan Rosenkilde), astorjoh@uwaterloo.ca (Arne Storjohann)

Both generalisations trace back to Hermite (1874) for the proof that e is transcendental, and were subsequently studied in greater detail by his student Padé (1892). There is a duality between the solution sets of the two problems, first observed by Mahler (1968). See also (Baker and Graves-Morris, 1996) 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. Barel and Bultheel (1992); Beckermann (1992); Beckermann and Labahn (1994). 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 $\boldsymbol{\lambda} \in \mathsf{K}[x]^{1\times t}$ and $\boldsymbol{\phi} \in \mathsf{K}[x]^{1\times n}$, such that $\boldsymbol{\lambda S} = \boldsymbol{\phi} \bmod x^d$. In matrix form:
>
> $$\begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_t \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1n} \\ S_{21} & S_{22} & \cdots & S_{2n} \\ \vdots & \vdots & & \vdots \\ S_{t1} & S_{t2} & \cdots & S_{tn} \end{bmatrix} \equiv \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_n \end{bmatrix} \bmod x^d \, . \qquad (1.1)$$

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 spectrum 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 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 $\tilde{O}(n^{\omega-1}td)$, where $O(n^\omega)$ is the cost of multiplying two square matrices over K of dimension at most n and $\tilde{O}(\cdot)$ ignores log-factors, see Section 2.1. Up to log-factors, this matches the previously best cost which uses a randomized algorithm, see below. Furthermore, our algorithms produce a parametrisation of *all* solutions, and it is unclear to us if this would be possible with the approach of the randomized algorithm.

Padé approximation has many applications in both numerical domains, e.g. control theory, and in symbolic algebraic domains, e.g. coding theory and cryptography. In particular, solving a classical Padé approximation is the core task in decoding Reed–Solomon codes (Berlekamp, 1968; Fitzpatrick, 1995), and the vector and matrix generalisations appear in list- and power decoding of Reed–Solomon codes, e.g. Roth and Ruckenstein (2000); Zeh et al. (2011); Schmidt et al. (2010); Rosenkilde (2018). Note that in Schmidt et al. (2010), the simultaneous Hermite–Padé approximation used for decoding has $1 = t \ll n$, while in Rosenkilde (2018) it has $1 < t < n$. A snag is that in both of these references, the degree restrictions that solutions to the Padé approximations need to satisfy are not *absolute* (as in Problem 1.1) but *relative*, i.e., of the form $\max_j(\deg \lambda_j + \mu_j) > \max_i(\deg \psi_i + \delta_i)$ for some $\mu_j, \delta_i \in \mathbb{Z}$, and we seek a solution which minimises the left-hand side of the inequality. Rosenkilde (2018) discusses how to leverage the parametrisation of all solutions to an absolute degree constraint problem, as produced by e.g. the algorithms of this paper, to find a single solution to the relative degree constraint problem.

Algorithms for the simultaneous Hermite–Padé approximation problem can be roughly split into two categories: the structured linear systems approach, and the approximant basis approach. Both approaches have roots in the extended euclidean algorithm. In the following discussion, we specialise the costs to the case $t < n$.

Imposing $\deg \lambda_i < T_i < d$ and $\deg \psi_j < N_j < d$ for some $T_i, N_j \in \mathbb{Z}$ and expanding Equation (1.1) to a linear system of homogeneous equations in the coefficients of $\lambda_1, \ldots, \lambda_t$ yields a system which is $t \times n$ block-Toeplitz with blocks of size $(d - N_j) \times T_i$. This structure was captured by the notion of displacement rank and solved in time $O(nt^2d^2)$ by Kailath et al. (1979), which was later improved to "super-fast" $O\tilde{~}(n^2td)$, see Bitmead and Anderson (1980); Morf (1980) for approaches assuming generic input and Kaltofen (1994) for a Las Vegas randomized algorithm that works for all inputs. See also Pan (2001) for the extensive history and many contributions in solving structured systems before 2000. Matrix-multiplication was integrated in the super-fast solvers for a cost of $O\tilde{~}(n^{\omega-1}td)$ by Bostan et al. (2008)[1], which was improved by logarithmic factors in Bostan et al. (2017); both are Las Vegas randomized. The latter is roughly a factor $\log(td)$ faster than the algorithms of this paper[2]. These algorithms typically return only a single solution; in particular, it is unclear to us if the algorithms of (Bostan et al., 2008, 2017) could be used to produce a parametrisation of all solutions at the same cost.

The approach of approximant basis traces back to especially Barel and Bultheel (1992) and Beckermann and Labahn (1994), and we discuss it in detail in Section 2.5.2. The congruence Equation (1.1) is reordered into a homogeneous system as $[\boldsymbol{\lambda} \mid \boldsymbol{\phi}]\boldsymbol{F} \equiv 0 \bmod x^d$, where

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

The set of vectors $\boldsymbol{v}$ such that $\boldsymbol{vF} \equiv 0 \bmod x^d$ is a $\mathsf{K}[x]$ module, a basis of which is known as a "minimal approximant basis of $\boldsymbol{F}$ to order d". Usually this equation is first solved modulo x using K-linear algebra and then lifted to higher powers of x using iterative cancellation or in a Newton iteration fashion. Barel and Bultheel (1992) solves this in complexity $O(n^3d^2)$. The first algorithm in Beckermann and Labahn (1994) gives the improved cost $O(n^2td^2)$ for our case, and their second algorithm incorporates fast polynomial multiplication to obtain an algorithm with complexity $O\tilde{~}(n^3d)$. This cost was improved by Giorgi et al. (2003) to $O\tilde{~}(n^\omega d)$. Up to logarithmic factors, this seems hard to improve in the case where the matrix $\boldsymbol{F}$ is roughly square. Handling wide matrices is easily done fast by sub-division, see Lemma 2.4, so attention therefore turned to the case where $\boldsymbol{F}$ is a tall rectangular matrix, where the cost was further improved in a series of papers (Zhou and Labahn, 2012; Jeannerod et al., 2016, In press); this applies to the simultaneous Hermite–Padé problem when $t > n$ for a cost of $O\tilde{~}(\max(t,n)^{\omega-1}nd)$.

However, for the case $t < n$, the matrix $\boldsymbol{F}$ is still roughly square and the previous best cost with the minimal approximant approach was still $O\tilde{~}(n^\omega d)$. The contribution of this paper is to compute simultaneous Hermite–Padé approximations by a more subtle use of minimal approximant bases, such that we can leverage the fast algorithms for the tall rectangular case, thereby improving the cost to $O\tilde{~}(n^{\omega-1}td)$.

Lastly, we will also mention the approach of row reduced matrices, which is closely related to approximant basis; we give more details in Section 2.5.1. The observation is that any vector $(\boldsymbol{\lambda} \mid \boldsymbol{\psi}) \in \mathsf{K}[x]^{t+n}$ in the row space of the matrix

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

[1] Note that in the earlier version of this paper (Rosenkilde né Nielsen and Storjohann, 2016), we erroneously claimed that the cost of applying Bostan et al. (2008) to the case $t = 1$ would cost $O\tilde{~}(n^\omega d)$.

[2] Before Problem 1.1 can be fed to a structured systems algorithm, one needs to compute from $\boldsymbol{S}$ appropriate "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 this is likely true.

will be a solution to the congruence Equation (1.1). Hence we are simply seeking small-degree vectors in the row space of this matrix, which can be found by computing a row reduced basis of $\boldsymbol{A}$. This approach is classical for the $\mathbb{Z}$-analogous problem of simultaneous Diophantine approximation, see e.g. (von zur Gathen and Gerhard, 2013, Chapter 17). For $\mathsf{K}[x]$, the earliest published reference we know of is (Olesh and Storjohann, 2007). If we use (Mulders and Storjohann, 2003) to perform the row reduction, and $t < n$, it was shown in (Nielsen, 2013) that this achieves the cost $O(n^2td^2)$, matching Beckermann and Labahn (1994) in a more general setting. Neiger (2016) gives the currently known fastest approach for performing row reduction which gives a cost $\tilde{O}(n^{\omega}d)$ when $t < n$. For the most general problem we consider, this is faster than the algorithms of this paper for certain ranges of parameters, see Section 2.5.1.

In the previous discussion, we assumed for simplicity that the moduli were all equal to x^d. We actually consider a generalisation where the moduli are replaced by arbitrary polynomials g_i. This generalises M-Padé problems as in e.g. (Barel and Bultheel, 1992), which posits that all g_i split over K. The minimal approximant approaches discussed support this generalisation at no extra asymptotic cost, see Section 2.5.2. The situation is a little more unclear for the structured linear system approach, but they could possibly be handled using the companion matrix displacement operator introduced in (Bostan et al., 2017).

The problem we study 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*

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

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

1. $(\lambda_1, \dots, \lambda_t)\boldsymbol{S}_i \equiv \phi_i \bmod g_i$ *for* $i = 1, \dots, n$*,*
2. *and* $\deg \lambda_j < T_j$ *for* $j = 1, \dots, t$ *and* $\deg \phi_i < N_i$ *for* $i = 1, \dots, n$*.*

We will call any vector $(\lambda_1, \dots, \lambda_t, \phi_1, \dots, \phi_n)$ as above *a solution* to the simultaneous Hermite–Padé approximation problem $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$. Note that if the entries of $\boldsymbol{N}$ are set too small, 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

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

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

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

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

Example 1.3. The following example demonstrates that the upper bound $T_i \leq \deg \operatorname{lcm}(g_1, \dots, g_n) + 1$ can be attained. Let $t = 1$, let K be a field with at least n elements, and let $g_i = x - a_i$ for pairwise different $a_1, \dots, a_n \in \mathsf{K}$. Let $T_1 = n + 1$ and $N_1 = \dots = N_n = 0$, and $\boldsymbol{S} = \mathbf{1}_{1 \times n}$. In other words, we seek a single $\lambda \in \mathsf{K}[x]$ such that $\lambda \equiv 0 \bmod (x - a_i)$ for $i = 1, \dots, n$, which only has the solution $\lambda = \prod_{i=1}^n g_i$ and multiples thereof.

On the other hand, the bound on T_i is always sufficient: notice that for any instance $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$, taking $\boldsymbol{\lambda} = (\operatorname{lcm}(g_1, \dots, g_n), 0, \dots, 0)$ yields $\boldsymbol{\lambda} \boldsymbol{S}_i \equiv 0 \bmod g_i$ for $i = 1, \dots, n$, satisfying any degree bounds $N_1, \dots, N_n$. ▲

A more ambitious goal than solving Problem 1.1 is to produce a basis which generates *all* solutions. Formally:

Problem 1.4 (($t \times n$) 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*[3]*.*

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 2.5 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. Even in the case all $g_i = x^d$, the number of field elements required to explicitly write down all of the entries of A could be $\Omega((t+n)^2 d)$. This means that any algorithm which produces or processes such a full basis A will not be able to attain our target cost bounds, e.g. the approaches of using row reduction or minimal approximant bases, as recalled in Section 2.5.

Instead, we observe that A is completely given by the problem instance as well as the first t columns of A, containing the λ_j polynomials.[4] Our algorithms will therefore represent A row-wise using the following compact representation.

Definition 1.5. For a given instance of Problem 1.4, a *solution specification* is a tuple $(\boldsymbol{\lambda}, \boldsymbol{\delta}) \in \mathsf{K}[x]^{k \times t} \times \mathbb{Z}^k_{<0}$ with $k \leq t + n$ and such that the *completion* of $\boldsymbol{\lambda}$ is a solution basis A, 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

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

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

[4] 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 uniquely reconstructed from its first t elements.

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

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

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 entry 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. ▲

Several earlier approaches which compute all solutions to a simultaneous Hermite–Padé approximation produce something similar to a $(-\boldsymbol{N})$-row reduced basis for all $(\boldsymbol{\lambda} \mid \boldsymbol{\phi}) \in \mathsf{K}[x]^{1\times(t+n)}$ which solve the congruences of Problem 1.1, including those which exceed the degree bound, e.g. (Barel and Bultheel, 1992; Beckermann and Labahn, 1992) and the one recalled in Section 2.5.1. As mentioned above, computing a full basis of solutions may exceed our target cost.

The take-away of this paper is the following general complexity for solving Problem 1.4:

Theorem 1.7. *There is an algorithm which can solve Problem 1.4 in complexity $\tilde{O}(\max(t,n)^{\omega-1} \min(t,n)d)$, where $d = \max_j T_j + \max_i \deg g_i$. When $t < n$, the algorithm outputs a solution specification $(\boldsymbol{\lambda}, \boldsymbol{\delta}) \in \mathsf{K}[x]^{k\times t} \times \mathbb{Z}^k_{<0}$, where $k \leq t + n$. When $t \geq n$, the algorithm outputs a complete solution basis $A \in \mathsf{K}[x]^{k\times(t+n)}$.*

The case $t \geq n$ is handled by either of the existing approaches recalled in Section 2.5, while the case $t < n$ is either of the two algorithms presented in this paper. See the theorems in the respective sections for a more precise description of the cost including log-factors. An immediate corollary is that we can compute the expanded form of one or a few solutions in the same cost:

Corollary 1.8. *There is an algorithm which solves Problem 1.1 in complexity $\tilde{O}(\max(t,n)^{\omega-1} \min(t,n)d)$, where $d = \max_j T_j + \max_i \deg g_i$.*

Both our algorithms depend crucially on recent developments on computing minimal approximant basis of matrices with fewer columns than rows (Zhou and Labahn, 2012; Jeannerod et al., 2016, In press). 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 2.5.2 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 (Gupta et al., 2012) and high-order lifting (Storjohann, 2003).

Our second algorithm in Section 4 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 in complexity

$\tilde{O}(t^\omega d)$ using the approaches recalled in Section 2.5. Two such solution bases 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 the λ-part. A solution basis of the full simultaneous Hermite–Padé problem is then obtained by structuring intersections along a binary tree.

We have decided to include both algorithms as we find that they both illuminate different facets of the simultaneous Hermite–Padé problem, but we cannot point to one being decidedly better than the other. The algorithms have comparable asymptotic complexity, and they both rely on an efficient computation of shifted minimal approximant bases. The first algorithm additionally requires high-order lifting for computing part of the inverse of a polynomial matrix. A more detailed asymptotic analysis including constants of leading-terms, or a carefully optimised implementation of the full algorithmic stack could point to which algorithm to prefer.

This paper is an extension of (Rosenkilde né Nielsen and Storjohann, 2016), where we considered only the simultaneous Padé problem, that is, an input of size $1 \times n$. Here we extend to the general case $t \times n$. In our previous paper 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.

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

2. Preliminaries

We begin by introducing our cost model, and then continue by gathering together some definitions and results regarding row reduced bases, minimal approximant bases, and their shifted variants.

2.1. Cost model

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

- $O(n^\omega)$ is the 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 $\omega < 2.373$ (Coppersmith and Winograd, 1990; Le Gall, 2014). 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 $\omega > 2$.

Cantor and Kaltofen (1991) show $\mathsf{M}(d) \in O(d \log(d) \operatorname{loglog}(d))$, while slightly better results are known for finite fields (Harvey et al., 2017). See also Harvey and van der Hoeven (2019) who show $\mathsf{M}(d) \in O(d \log(d))$ operations in K under an unproven number theoretic assumption. 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 $\mathsf{M}(d) \in O(d^{\omega-1-\epsilon})$; the purpose of this assumption is to ensure that if fast matrix multiplication techniques are used then fast polynomial multiplication should also be used. For example, in one of our cost analyses 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: $\mathsf{PM}(n,d) + \mathsf{PM}(n,d') \le \mathsf{PM}(n,d+d')$ for all $n,d,d' \ge 1$. 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 Cantor and Kaltofen (1991):

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

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 $\mathsf{PM}(c_1 n, c_2 d) \in O(\mathsf{PM}(n,d))$.

In our main theorems, we report complexities using the cost function PM and including logarithmic factors, but in discussions we often employ $\tilde{O}(\cdot)$ which omits logarithmic factors for simplicity.

2.2. *Degrees and shifted degrees*

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 $\mathrm{Row}(A)$ the $\mathsf{K}[x]$-linear row space of A.

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 degree* 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 degree* 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 the shifted variants of the notion of degree, row degree, and leading matrix (Barel and Bultheel, 1992; Zhou and Labahn, 2012; Jeannerod et al., 2016). For a *shift* $\boldsymbol{s} = (s_1,\ldots,s_m) \in \mathbb{Z}^m$, 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 $\deg_{\boldsymbol{s}} v := \deg v x^{\boldsymbol{s}}$, $\operatorname{rowdeg}_{\boldsymbol{s}} A := \operatorname{rowdeg} A x^{\boldsymbol{s}}$, and $\mathrm{LM}_{\boldsymbol{s}}(A) := \mathrm{LM}(A x^{\boldsymbol{s}})$. For a shift $t \in \mathbb{Z}^n$ we similarly have the $\boldsymbol{t}$-column degree $\operatorname{coldeg}_{\boldsymbol{t}} A := \operatorname{coldeg} x^{\boldsymbol{t}} A = \operatorname{rowdeg}_{\boldsymbol{t}} A^\top$. 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 by Jeannerod et al. (2016), up to negation the definition of $\boldsymbol{s}$-degree is equivalent to that used by Beckermann et al. (2006) and to the notion of *defect* in Beckermann and Labahn (1994).

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

2.3. *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 is 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 (Kailath, 1980, Section 6.3.2) and (Zhou, 2012, Section 2.7))**.** *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*

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

Property 3 in Theorem 2.1 is known as the "predictable degree"-property (Kailath, 1980, 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 matrices, 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, see e.g. (Zhou, 2012, Lemmas 2.19 and 2.20) for a proof:[5]

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 $\boldsymbol{r} = \mathrm{rowdeg}_{\boldsymbol{s}} F_1$. Then F_2F_1 is $\boldsymbol{s}$-row reduced with $\mathrm{rowdeg}_{\boldsymbol{s}}(F_2F_1) = \mathrm{rowdeg}_{\boldsymbol{r}}(F_2)$.*

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 the one of Beckermann et al. (1999):

Definition 2.3. A non-singular matrix $R \in \mathsf{K}[x]^{n\times n}$ is in $\boldsymbol{s}$-Popov form if $\mathrm{LM}_{\boldsymbol{s}}(R)$ 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.

2.4. Approximant and minimal approximant basis

We recall the standard notion of (left) minimal approximant basis, sometimes known as order basis or σ-basis (Beckermann and Labahn, 1994). For a matrix $A \in \mathsf{K}[x]^{n\times m}$ and order $d \in \mathbb{Z}_{\geq 0}$, an *order d (left) approximant* is a vector $\boldsymbol{p} \in \mathsf{K}[x]^{1\times n}$ such that $\boldsymbol{p}A \equiv \boldsymbol{0} \bmod x^d$.

A (left) *approximant basis of order d* is 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 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 $A\boldsymbol{p} \equiv \boldsymbol{0} \bmod x^d$, 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 $\boldsymbol{\delta} = \mathrm{rowdeg}_{\boldsymbol{s}} F$. Note that F is not canonical, and we allow

[5] (Zhou, 2012, Lemma 2.20) has a typo: the last part of should read "if and only if $\boldsymbol{AB}$ is $\boldsymbol{u}$-column reduced".

$\mathsf{MinBasis}$ to return any such $\boldsymbol{s}$-minimal approximant basis. It follows from Theorem 2.1 that $\boldsymbol{\delta}$ will be the same for all of these up to ordering of the entries.

The next lemma recalls a well known structural recursion for minimal approximant bases, which traces back to (Beckermann and Labahn, 1997, Theorem 5.1). 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* $A = \left[A_1 \mid A_2\right]$ *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 the function $\mathsf{NegMinBasis}(d, A, \boldsymbol{s})$ have the same output as $\mathsf{MinBasis}$, but with F restricted to the negative part.

We will use the following easy statement on the determinant of minimal approximant bases:

Lemma 2.5. *Let* $F \in \mathsf{K}[x]^{n \times n}$ *be an approximant basis of order* d *for some* $A \in \mathsf{K}[x]^{n \times m}$. *Then* $\det F \mid x^{dm}$.

Proof. Note that there exists a unimodular matrix $U \in \mathsf{K}[x]^{n \times n}$ such that the last $n - m$ rows of UA are zero. Then the matrix $\bar{U}$ obtained from U by multiplying the first m rows by x^d will satisfy $\bar{U} A \equiv 0 \bmod x^d$. So the row space of $\bar{U}$ is contained in the row space of F, which implies $\det F \mid \det \bar{U}$, and the latter is x^{md} up to a constant. □

Many problems of $\mathsf{K}[x]$ matrices or approximations reduce to the computation of (shifted) minimal approximant bases, see e.g. Beckermann and Labahn (1994) and Giorgi et al. (2003), often resulting in the best known asymptotic complexities for these problems. Part 1 of the following theorem is a special case of (Jeannerod et al., In press, Theorem 1.1). Part 2 is (Jeannerod et al., 2017, 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 $\mathsf{PopovMinBasis}$, which computes a right minimal approximant basis in shifted column-Popov form.

Note that (Jeannerod et al., In press) 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.

2.5. Existing algorithms for simultaneous Hermite–Padé approximations

Let $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$ be an instance of Problem 1.4 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.

2.5.1. 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 a $(-\boldsymbol{N})$-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 forms a solution basis. Therefore, if $\boldsymbol{\lambda}$ is the matrix consisting of the first t columns of this sub-matrix and $\boldsymbol{\delta}$ the $(-\boldsymbol{N})$-row degree 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.

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 (Sugiyama et al., 1976; Justesen, 1976; Gustavson and Yun, 1979). The complexity of this is $O(\mathsf{M}(\deg g_1)\log(\deg g_1))$.

For $t < n$ and if $k = \deg \boldsymbol{g}$ then one can use (Neiger, 2016) to compute the $(-\boldsymbol{N})$-shifted Popov form of A at the cost $\tilde{O}(n^{\omega}k)$. Note that this cost holds even when $k \ll T_j \leq \deg \operatorname{lcm}(g_1, \ldots, g_n)+1$. For this case, the algorithms of our paper have complexity up to $\tilde{O}(n^{\omega}tk)$.

Another approach is to use the iterative row-reduction algorithm of Mulders and Storjohann (2003): if $T_j \in O(\max_i(\deg g_i))$ for each $j = 1, \ldots, t$ (e.g. if each $g_i = x^{d_i}$), then the analysis of (Nielsen, 2013) shows that this approach will cost $O(n^2tk^2)$; this matches the first algorithm of Beckermann and Labahn (1994) but for a more general set of simultaneous Hermite Padé approximations.

2.5.2. 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 $\boldsymbol{\lambda}\boldsymbol{S}_i \equiv \phi_i \bmod x^d$ 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

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

where 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 1.4 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 is exactly the set of 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 $\boldsymbol{\lambda}\boldsymbol{S}_i \equiv \phi_i + q_i g_i \bmod x^d$ 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:

Algorithm 1 DirectSHPade

```
Input: $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$, an instance of Problem 1.4 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 $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$.
    $\boldsymbol{h} \leftarrow -(\boldsymbol{N} \mid T-1, \dots, T-1) \in \mathbb{Z}^{2n+t}$, where $T = \max_i T_i$
    $d \leftarrow T + \max_i \deg g_i - 1$
    $B = \begin{bmatrix} -\boldsymbol{S} \\ I_{n\times n} \\ \mathrm{diag}(\boldsymbol{g}) \end{bmatrix}$
    $(\begin{bmatrix} \boldsymbol{\lambda} \mid * \end{bmatrix}, \boldsymbol{\delta}) \leftarrow$ NegMinBasis$(d, B, \boldsymbol{h})$
    return $(\boldsymbol{\lambda}, \boldsymbol{\delta})$
```

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

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

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

3. Algorithm 1: Reduction to the Dual

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

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

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 by Mahler (1968) in a special case, and later more generally (Beckermann and Labahn, 1992, 1997), was previously exploited in (Beckermann and Labahn, 2009) to develop algorithms for the fraction-free computation of simultaneous Padé approximations.

3.1. Duals of minimal approximant bases

We begin by developing a general theory of minimal approximant basis “duality”, and how to perform the computations efficiently.

For a nonsingular $A \in \mathsf{K}[x]^{n\times n}$ recall that the adjoint of A, denoted by $\mathrm{adj}(A)$, is equal to $(\det A)A^{-1}$, and that entry $\mathrm{adj}(A)_{j,i}$ 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, the entries of $\mathrm{adj}(A)$ are in $\mathsf{K}[x]$.

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 $(Ax^{\boldsymbol{s}})\operatorname{adj}(Ax^{\boldsymbol{s}}) = (\det Ax^{\boldsymbol{s}})I_{m\times m}$. 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 $\eta - \eta_i$. Therefore, the row-wise leading coefficient matrix of $Ax^{\boldsymbol{s}}$ multiplied with the column-wise leading coefficient matrix of $\operatorname{adj}(Ax^{\boldsymbol{s}})$ is the identity matrix up to K-scaling, and 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 $AB = x^d I_{n\times n}$. 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 $k \le nd$ 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^dA^{-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^dB^{-1} is a polynomial matrix.*

Proof. The rows of $x^dI_{n\times n}$ are all approximants of order d, so they are contained in the row space of A. Therefore there is a $B \in \mathsf{K}[x]^{n\times n}$ such that $BA = x^dI_{n\times n}$, and hence $x^dA^{-1} = B$. 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 $B = x^dA^{-1}$ has entries in $\mathsf{K}[x]$ with

$$AB = x^dI_{n\times n} \qquad B = x^dA^{-1} \qquad A = x^dB^{-1}. \tag{3.1}$$

Symmetrically, if B is as in Lemma 3.3, then $A = x^dB$ has entries in $\mathsf{K}[x]$ and (3.1) also holds. Every left (right) minimal approximant basis thus has a natural right (left) dual basis. Note that $d \le \deg\det A$, and is often much smaller, so Lemma 3.3 shows that a much smaller multiple of A^{-1} brings it into $\mathsf{K}[x]$ than the adjoint $\operatorname{adj}(A) = \det A \cdot A^{-1}$.

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

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

$$x^d I_{n\times n} = AB = AGG^{-1}B = Rx^d G^{-1}B\,.$$

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 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 $\mathrm{adj}(G)$ is $(-\boldsymbol{s})$-row reduced with

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

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

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

□

Suppose a nonsingular $B \in \mathsf{K}[x]^{n\times n}$ enjoys the property that $x^d B^{-1}$ has entries in $\mathsf{K}[x]$. 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 $m \leq n$. 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 $md < n$.*

The proof of this theorem 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. $\sum \text{coldeg} F \leq md$.
2. $x^d F^{-1}$ *has entries in* $\mathsf{K}[x]$.
3. $\deg x^d F^{-1} \leq d$.

Proof. Part 1 follows from Lemma 2.5 after noting that F is column reduced since it is in shifted (row-wise) Popov form and hence $\sum \text{coldeg} F = \deg \det F$.

Part 2 is Lemma 3.3.

For Part 3, consider F and $x^d F^{-1}$ as polynomials with matrix coefficients. Since F is column reduced its leading coefficient is full rank, and hence $d = \deg(x^d F^{-1} F) = \deg(x^d F^{-1}) + \deg F$. □

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 (Storjohann, 2003, 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 follows from (Gupta et al., 2012, 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 2 and 3) 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 2.5), 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 expansion of (3.2) to a desired precision $p \in \mathbb{Z}_{\geq 0}$. Since $\deg x^d F^{-1} \leq d$ (Lemma 3.6.3), 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 (Storjohann, 2003, Proposition 15) the lifting has cost

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

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 (von zur Gathen and Gerhard, 2013, 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 $md < n$, 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 3.6.1; 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: $\boldsymbol{v}F^{-1} = \left[\begin{array}{c|c} \boldsymbol{v}_1F_1^{-1} & \boldsymbol{v}_2 \end{array}\right]P$. As in the proof of Lemma 3.8, we will now use high-order lifting combined with partial linearisation to compute $\boldsymbol{v}_1F_1^{-1}$.

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 $O((\log d)m^2d\,\mathsf{M}(d))$. Using the assumption $\mathsf{M}(d) \in O(d^{\omega-1})$ we see that $m^2d\,\mathsf{M}(d) \in O((md)^{\omega})$. □

3.3. The dual of a simultaneous Hermite–Padé problem

Theorem 3.10. *Let $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$ be an 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^dG^{-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 $\mathrm{rowdeg}_{(-\boldsymbol{s})}(x^dG^{-1}) = (d-\eta_1,\ldots,d-\eta_{2n+t})$.

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

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

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 $\hat{A}\hat{B} = x^dI_{(2n+t)\times(2n+t)}$, 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:

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

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*

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

For $i = 1, \dots, n$ 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 $\boldsymbol{h}_i := \operatorname{coldeg}_{\boldsymbol{s}_i} H_i$. Then a right minimal approximant basis of the matrix $[0_{n\times n} \mid \operatorname{diag}(\boldsymbol{g}) \mid I_{n\times n}]$ is given by G_1 where

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

with

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

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 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.

Algorithm 2 DualitySimPade

```
Input: $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$, an instance of Problem 1.4 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 $T \leftarrow \max_i T_i$
 2 $d \leftarrow T + \max_i \deg g_i - 1$
 3 $(H_i, \boldsymbol{h}_i) \leftarrow \mathsf{PopovMinBasis}_{\mathsf{Right}}(d, [-g_i\ 1], (N_i, T-1))$ for $i = 1, \dots, n$
 4 $(G_1, \boldsymbol{h}) \leftarrow$ as in (3.6) with $P$ as in (3.5)
 5 $A_2 \leftarrow [I_{t\times t}\ \ \boldsymbol{S}\ \ \boldsymbol{0}_{t\times n}] \in \mathsf{K}[x]^{t\times(2n+t)}$
 6 $(G_2, \boldsymbol{\eta}) \leftarrow \mathsf{PopovMinBasis}_{\mathsf{Right}}(d, A_2G_1, \boldsymbol{h})$
 7 $\hat{\boldsymbol{\lambda}} \leftarrow$ first $t$ columns of $x^dG_2^{-1}$
 8 $\hat{\boldsymbol{\delta}} \leftarrow (d - \eta_1, \dots, d - \eta_{n+1})$
 9 $I \leftarrow \{i \mid \hat{\delta}_i < 0\}$, and $k \leftarrow |I|$
10 $(\boldsymbol{\lambda},\ \boldsymbol{\delta}) \leftarrow (\hat{\boldsymbol{\lambda}}_{i\in I},\ (\hat{\boldsymbol{\delta}}_i)_{i\in I}) \in \mathsf{K}[x]^{k\times t} \times \mathbb{Z}^k$
11 return $(\boldsymbol{\lambda}, \boldsymbol{\delta})$
```

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 $\boldsymbol{s} = (\boldsymbol{N} \mid T-1, \dots, T-1)$ and d be as in the algorithm. By combining Theorem 2.7 and Theorem 3.10, if $(F, \boldsymbol{f}) = \mathsf{PopovMinBasis}_{\mathsf{Right}}(d, A, \boldsymbol{s})$, the submatrix of x^dF^{-1} comprised of those rows with negative $(-\boldsymbol{s})$-degree forms a solution specification to the simultaneous Hermite–Padé approximation. By combining Lemma 2.4 and Lemma 3.11, then G_1G_2 is a right $\boldsymbol{s}$-minimal approximant basis of A with $\mathrm{coldeg}_{\boldsymbol{s}}(G_1G_2) = \boldsymbol{\eta}$. 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 $[I_{t\times t} \mid \boldsymbol{0}]^\top$, so the first t columns of $x^d(G_1G_2)^{-1}$ are just the first t columns of $x^dG_2^{-1}$, as assigned to $\hat{\boldsymbol{\lambda}}$ in Line 7. By Theorem 3.10 then $\hat{\boldsymbol{\delta}}$ is the $(-\boldsymbol{s})$-row degree 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 $G_1 G_2$, where $(G_1, \boldsymbol{h}) = \mathsf{PopovMinBasis}_{\mathsf{Right}}(d, A_2, \boldsymbol{s})$, and $(G_2, \boldsymbol{\eta}) = \mathsf{PopovMinBasis}_{\mathsf{Right}}(d, A_1 G_1, \boldsymbol{h})$. Such G_1 and G_2 are given by

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

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

$$\boldsymbol{\eta} = \mathrm{coldeg}_{\boldsymbol{s}} G_1 G_2 = \mathrm{coldeg}_{\boldsymbol{h}} G_2 = (10,\ 9,\ 9,\ 9,\ 9,\ 10,\ 9,\ 9,\ 8,\ 9)\ .$$

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

$$\begin{bmatrix} (x^d F^{-1})_1 \\ \hline (x^d F^{-1})_6 \end{bmatrix} \trianglelefteq \begin{bmatrix} 4 & 2 & 0 & 2 & 3 & 1 & 3 & 3 & 3 & 3 \\ 3 & 1 & 0 & 1 & 1 & 3 & 2 & 2 & 2 & 2 \end{bmatrix} .$$

$x^d F^{-1}$ 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 bases 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

Before defining our notion of t-intersection, let us first discuss the simpler case of row space intersections. 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; conversely, for any vector $\boldsymbol{v}$ in both the row space of F_1 and F_2, there are $\boldsymbol{b}_1, \boldsymbol{b}_2$ such that $(\boldsymbol{v} \mid \boldsymbol{b}_1 \mid \boldsymbol{b}_2)$ is in the left kernel of R.

Suppose now that we seek only *small* vectors in the intersection, say $\deg \boldsymbol{v} < d$ with d also bounding the degree of F_1 and F_2, and suppose that both F_1 and F_2 are row reduced. Instead of computing the kernel of R, it is cheaper to compute a left minimal approximant basis M of R of order $2d$. The kernel of R is of course contained in the row space of M, but conversely, if $\boldsymbol{m}R \equiv 0 \bmod x^{2d}$ with $\deg \boldsymbol{m} < d$, then $\deg(\boldsymbol{m}R) < 2d$ so the congruence lifts to an equality, and $\boldsymbol{m}$ must be a kernel vector. Hence, the first m columns of $\mathsf{NegMinBasis}(d, R, (-d, \ldots, -d))$ is row reduced and generates all the small vectors in $\mathrm{Row}(F_1) \cap \mathrm{Row}(F_2)$.

Consider now that F_1 is the first part of a larger matrix $A_1 = [F_1 \mid H_1]$ and similarly $A_2 = [F_2 \mid H_2]$. 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. This motivates the following generalisation of row space intersections:

Definition 4.1. Let $A_1 = [\lambda_1 \mid H_1] \in \mathsf{K}[x]^{k_1 \times (t+n_1)}$ and $A_2 = [\lambda_2 \mid H_2] \in \mathsf{K}[x]^{k_2 \times (t+n_2)}$. The *t-intersection* of A_1 and A_2 is the $\mathsf{K}[x]$-module:

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

Consider shifts $\boldsymbol{h}_1 = (\boldsymbol{v} \mid \boldsymbol{s}_1) \in \mathbb{Z}^{t+n_1}$ and $\boldsymbol{h}_2 = (\boldsymbol{v} \mid \boldsymbol{s}_2) \in \mathbb{Z}^{t+n_2}$ sharing the first t components. If A_i is $\boldsymbol{h}_i$-row reduced for $i = 1, 2$, 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 $A_1 = [\lambda_1 \mid H_1] \in \mathsf{K}[x]^{k_1 \times (t+n_1)}$ and $A_2 = [\lambda_2 \mid H_2] \in \mathsf{K}[x]^{k_2 \times (t+n_2)}$ be $\boldsymbol{h}_1$-resp. $\boldsymbol{h}_2$-row reduced.*

Let $\boldsymbol{r} = (\boldsymbol{v} \mid \mathrm{rowdeg}_{\boldsymbol{h}_1}(H_1) \mid \mathrm{rowdeg}_{\boldsymbol{h}_2}(H_2)) \in \mathbb{Z}^{t+k_1+k_2}$ and let M be an $\boldsymbol{r}$-row reduced kernel basis of R, where

$$R = \left[\begin{array}{c|c} I_{t\times t} & I_{t\times t} \\ \hline -\lambda_1 & \\ & -\lambda_2 \end{array}\right] .$$

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 $\operatorname{rowdeg}_{\boldsymbol{h}}(MC) = \operatorname{rowdeg}_{\boldsymbol{r}}(M)$*, where* $\boldsymbol{h} = (\boldsymbol{v} \mid \boldsymbol{s}_1 \mid \boldsymbol{s}_2)$*. In particular, the first* t *columns of* M *are 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 sub-matrix consisting of the first t columns of M, then $\operatorname{Row}(M')$ is the set of vectors that lie in both $\operatorname{Row}(\lambda_1)$ and $\operatorname{Row}(\lambda_2)$. 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 $\operatorname{rowdeg}_{\boldsymbol{h}}(MC) = \operatorname{rowdeg}_{\boldsymbol{r}}(M)$, consider the following amalgamation of R and C:

$$F = \left[\begin{array}{ccccc} I_{t\times t} & I_{t\times t} & & I_{t\times t} & \\ \hline & -\lambda_1 & -H_1 & & \\ & & & -\lambda_2 & -H_2 \end{array}\right].$$

Since $[\lambda_1 \mid H_1]$ and $[\lambda_2 \mid H_2]$ 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 $\boldsymbol{r} = \operatorname{rowdeg}_{\boldsymbol{h}'}(F)$. Thus by Lemma 2.2, MF is $\boldsymbol{h}'$-row reduced with $\operatorname{rowdeg}_{\boldsymbol{h}'}(MF) = \operatorname{rowdeg}_{\boldsymbol{r}}(M)$. But since M is a kernel basis of R, the matrix 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 $\operatorname{rowdeg}_{\boldsymbol{h}}(MC) = \operatorname{rowdeg}_{\boldsymbol{r}}(M)$. □

Since R of Theorem 4.2 has rank at least t, 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 only need to 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 $\boldsymbol{v} = (\boldsymbol{\lambda} \mid \boldsymbol{a}_1 \mid \boldsymbol{a}_2)$. Either $(\boldsymbol{\lambda} \mid a_1) \notin \operatorname{Row}(B_1)$ or $(\boldsymbol{\lambda} \mid a_2) \notin \operatorname{Row}(B_2)$; assume without loss of generality 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 2.5.1, if $P \in \mathsf{K}[x]^{(t+2n)\times(t+2n)}$ is a $(-\boldsymbol{N})$-row reduced basis of A, where

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

then the sub-matrix of P comprised of the rows with negative $(-\boldsymbol{N})$-degree is 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 $(\boldsymbol{\lambda}, \boldsymbol{a}_i) \in \operatorname{Row}(A_i), i = 1, 2$, 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 suffices 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 $\mathsf{NegMinBasis}$. 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. ▲

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

Algorithm 3 RecursiveSHPade

```
Input: $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$, an instance of Problem 1.4 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     $(T_1, \dots, T_t, N_1, \dots, N_n) \leftarrow \boldsymbol{N}$
 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     $\boldsymbol{N}_1 \leftarrow (T_1, \dots, T_t, N_1, \dots, N_{\lceil n/2 \rceil})$
 8     $\boldsymbol{N}_2 \leftarrow (T_1, \dots, T_t, N_{\lceil n/2 \rceil + 1}, \dots, N_n)$
 9     $(\boldsymbol{\lambda}_1, \boldsymbol{\delta}_1) \leftarrow$ RecursiveSHPade$(\boldsymbol{S}_1, \boldsymbol{g}_1, \boldsymbol{N}_1)$
10     $(\boldsymbol{\lambda}_2, \boldsymbol{\delta}_2) \leftarrow$ RecursiveSHPade$(\boldsymbol{S}_2, \boldsymbol{g}_2, \boldsymbol{N}_2)$
11     $\boldsymbol{r} \leftarrow (-T_1, \dots, -T_t \mid \boldsymbol{\delta}_1 \mid \boldsymbol{\delta}_2)$
12     $R \leftarrow \left[\begin{array}{c|c} I_{t\times t} & I_{t\times t} \\ \hline -\boldsymbol{\lambda}_1 & \\ \hline & -\boldsymbol{\lambda}_2 \end{array}\right]$
13     $(\left[\begin{array}{c|c} \boldsymbol{\lambda} & * \end{array}\right], \boldsymbol{\delta}) \leftarrow$ 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
```

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 $\operatorname{rowdeg}_{-\boldsymbol{N}_i}(P_i') = \boldsymbol{\delta}_i$. Note that P_i' is the negative part of some $(-\boldsymbol{N}_i)$-row reduced matrix P_i which is row-equivalent to

$$A_i = \left[\begin{array}{c|c} I_{t\times t} & \boldsymbol{S}_i \\ \hline & \operatorname{diag}(\boldsymbol{g}_i) \end{array}\right],$$

as according to Section 2.5. By the induction hypothesis and from the discussion at the beginning of the section, 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 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 $\operatorname{rowdeg}_{(-T,\dots,-T)} \boldsymbol{\lambda}_i \leq \boldsymbol{\delta}_i$ 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 $\boldsymbol{w}R \equiv 0 \bmod x^T$ 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 $T := \max\{T_1, \dots, T_t\}$. The call to NegMinBasis in Line 13 uses an order bounded by T, but for simplicity we will use the upper bound $d := T + \max\deg g_i$. We get the following recursion on $C(n)$:

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

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 $n < td$, 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:

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

If $n \geq td$, then let $k \in \Theta(\log(n/(td)))$ be the largest integer such that $n/2^k \geq td$. 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 that 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

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

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 λ_1 and λ_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 such a specification 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.

References

G.A. Baker and P.R. Graves-Morris. *Padé approximants*, volume 59. Cambridge Univ. Press, 1996.

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

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.

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

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

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

Bernhard Beckermann and George Labahn. Recursiveness in matrix rational interpolation problems. *J. Comp. App. Math.*, 77(1–2):5–34, January 1997.

Bernhard Beckermann and George Labahn. Fraction-Free Computation of Simultaneous Padé Approximants. In *Proc. of ISSAC*, pages 15–22, 2009.

Bernhard Beckermann, George Labahn, and Gilles Villard. Shifted Normal Forms of Polynomial Matrices. In *Proc. of ISSAC*, pages 189–196. ACM, 1999. ISBN 1-58113-073-2.

Elwyn R. Berlekamp. *Algebraic Coding Theory*. Aegean Park Press, 1968.

R. R. Bitmead and Brian D. O. Anderson. Asymptotically Fast Solution of Toeplitz and Related Systems of Linear Equations. *Lin. Alg. Appl.*, 34(DEC):103–116, 1980. doi: 10.1016/0024-3795(80)90161-5.

Alin Bostan, Claude-Pierre Jeannerod, and Eric Schost. Solving structured linear systems with large displacement rank. *Th. Comp. Sc.*, 407(1–3):155–181, November 2008.

Alin Bostan, C.-P. Jeannerod, Christophe Mouilleron, and É Schost. On matrices with displacement structure: Generalized operators and faster algorithms. *SIAM J. Matr. Anal. Appl.*, 38(3):733–775, 2017.

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

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

P. Fitzpatrick. On the Key Equation. *IEEE Trans. Inf. Theory*, 41(5):1290–1302, 1995.

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

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. doi: 10.1016/j.jsc.2011.09.006.

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. doi: 10.1109/TCS.1979.1084696.

David Harvey and Joris van der Hoeven. Polynomial multiplication over finite fields in time O (n log n). *Preprint*, 2019.

David Harvey, Joris van der Hoeven, and Grégoire Lecerf. Faster polynomial multiplication over finite fields. *Journal of the ACM*, 63(6), February 2017.

Charles Hermite. *Sur la fonction exponentielle*. Gauthier-Villars, 1874.

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

Claude-Pierre Jeannerod, Vincent Neiger, Éric Schost, and Gilles Villard. Fast computation of minimal interpolation bases in popov form for arbitrary shifts. In *Proc. of ISSAC*, pages 295–302, 2016. ISBN 978-1-4503-4380-0. doi: 10.1145/2930889.2930928.

Claude-Pierre Jeannerod, Vincent Neiger, and Gilles Villard. Fast computation of approximant bases in canonical form. *J. Symb Comp.*, In press. doi: 10.1016/j.jsc.2019.07.011.

J. Justesen. On the complexity of decoding Reed-Solomon codes (Corresp.). *IEEE Trans. Inf. Theory*, 22(2):237–238, March 1976. doi: 10.1109/TIT.1976.1055516.

T Kailath. *Linear Systems*. Prentice-Hall, 1980.

Thomas Kailath, Sun-Yuan Kung, and Martin Morf. Displacement ranks of matrices and linear equations. *Journal of Mathematical Analysis and Applications*, 68(2):395–407, April 1979. doi: 10.1016/0022-247X(79)90124-0.

Erich Kaltofen. Asymptotically fast solution of Toeplitz-like singular linear systems. In *Proc. of ISSAC*, pages 297–304, 1994.

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

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

Martin Morf. Doubling algorithms for Toeplitz and related equations. In *Proc. of IEEE ICASSP*, volume 5, pages 954–959, 1980.

T. Mulders and A. Storjohann. On Lattice Reduction for Polynomial Matrices. *J. Symb. Comp.*, 35(4):377–401, 2003.

Vincent Neiger. Fast computation of shifted Popov forms of polynomial matrices via systems of modular polynomial equations. In *Proc. of ISSAC*, July 2016. doi: 10.1145/2930889.2930936.

Johan S. R. Nielsen. Generalised Multi-sequence Shift-Register Synthesis using Module Minimisation. In *Proc. of IEEE ISIT*, 2013.

Zach Olesh and Arne Storjohann. The vector rational function reconstruction problem. In *Computer Algebra 2006: Latest Advances in Symbolic Algorithms*, pages 137–149. World Scientific, 2007.

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

Victor Y Pan. *Structured matrices and polynomials: unified superfast algorithms*. Birkhäuser, 2001.

Johan Rosenkilde. Power Decoding of Reed–Solomon Up to the Johnson Radius. *Advances in Mathematics of Communications*, 12(1):81–106, February 2018. doi: 10.3934/amc.2018005.

Johan Rosenkilde né Nielsen and Arne Storjohann. Algorithms for Simultaneous Padé Approximations. In *Proc. of ISSAC*, pages 405–412, 2016. ISBN 978-1-4503-4380-0. doi: 10.1145/2930889.2930933.

R.M. Roth and G. Ruckenstein. Efficient Decoding of Reed–Solomon Codes Beyond Half the Minimum Distance. *IEEE Trans. Inf. Theory*, 46(1):246 –257, 2000.

G. Schmidt, V.R. Sidorenko, and M. Bossert. Syndrome Decoding of Reed-Solomon Codes Beyond Half the Minimum Distance Based on Shift-Register Synthesis. *IEEE Trans. Inf. Theory*, 56(10):5245–5252, 2010.

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

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

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.

J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge Univ. Press, 3rd edition, 2013.

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.

Wei Zhou. *Fast Order Basis and Kernel Basis Computation and Related Problems*. PhD thesis, University of Waterloo, 2012.

Wei Zhou and George Labahn. Efficient algorithms for order basis computation. *J. Symb. Comp.*, 47(7):793–819, 2012.