Algorithms for Simultaneous Padé Approximations

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

Arne Storjohann
University of Waterloo
Canada
astorjoh@uwaterloo.ca

ABSTRACT

We describe how to solve simultaneous Padé approximations over a power series ring $\mathsf{K}[[x]]$ for a field K using $O^{\sim}(n^{\omega-1}d)$ operations in K, where d is the sought precision and n is the number of power series to approximate. We develop two algorithms using different approaches. Both algorithms return a reduced sub-bases that generates the complete set of solutions to the input approximations problem that satisfy the given degree constraints. Our results are made possible by recent breakthroughs in fast computations of minimal approximant bases and Hermite Padé approximations.

1. INTRODUCTION

The Simultaneous Padé approximation problem concerns approximating several power series $S_1, \ldots, S_n \in \mathsf{K}[[x]]$ with rational functions $\frac{\sigma_1}{\lambda}, \ldots, \frac{\sigma_n}{\lambda}$, all sharing the same denominator λ. In other words, for some $d \in \mathbb{Z}_{\geq 0}$, we seek $\lambda \in \mathsf{K}[x]$ of low degree such that each of

$$\operatorname{rem}(\lambda S_1,\ x^d), \operatorname{rem}(\lambda S_2,\ x^d),\ \ldots,\ \operatorname{rem}(\lambda S_n,\ x^d)$$

has low degree. The study of Simultaneous Padé approximations traces back to Hermite's proof of the transcendence of e [18]. Solving Simultaneous Padé approximations has numerous applications, such as in coding theory, e.g. [13, 28]; or in distributed, reliable computation [11]. Many algorithms have been developed for this problem, see e.g. [3, 26, 27, 29] as well as the references therein. Usually one cares about the regime where $d \gg n$. Obtaining $O(nd^2)$ is classical through successive cancellation, see [4] or [13] for a Berlekamp–Massey-type variant. Using fast arithmetic, the previous best was $O^{\sim}(n^{\omega}d)$, where ω is the exponent for matrix multiplication, see Section 1.1. That can be done by computing a minimal approximant basis with e.g. [15, 16]; this approach traces back to [2, 3]. Another possibility which achieves the same complexity is fast algorithms for solving structured linear systems, e.g. [8]; see [10] for a discussion of this approach.

A common description is to require $\deg \lambda < N_0$ for some degree bound N_0, and similarly $\deg \operatorname{rem}(\lambda S_1,\ x^d) < N_i$ for $i = 1, \ldots, n$. The degree bounds could arise naturally from the application, or could be set such that a solution must exist. A natural generalisation is also to replace the x^d moduli with arbitrary $g_1, \ldots, g_n \in \mathsf{K}[x]$. Formally, for any field K:

PROBLEM 1. *Given a tuple $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$ where*

- *$\boldsymbol{S} = (S_1, \ldots, S_n) \in \mathsf{K}[x]^n$ is a sequence of polynomials,*
- *$\boldsymbol{g} = (g_1, \ldots, g_n) \in \mathsf{K}[x]^n$ is a sequence of moduli polynomials with $\deg S_i < \deg g_i$ for $i = 1, \ldots, n$,*
- *and $\boldsymbol{N} = (N_0, \ldots, N_n) \in \mathbb{Z}_{\geq 0}^{n+1}$ are degree bounds satisfying $1 \leq N_0 \leq \max_i \deg g_i$ and $N_i \leq \deg g_i$ for $i = 1, \ldots, n$,*

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

1. *$\lambda S_i \equiv \phi_i \mod g_i$ for $i = 1, \ldots, n$, and*
2. *$\deg \lambda < N_0$ and $\deg \phi_i < N_i$ for $i = 1, \ldots, n$.*

We will call any vector $(\lambda, \phi_1, \ldots, \phi_n)$ as above *a solution* to a given Simultaneous Padé approximation problem. Note that if the N_i are set too low, then it might be the case that no solution exists.

EXAMPLE 2. *Consider over $\mathbb{F}_2[x]$ that $g_1 = g_2 = g_3 = x^5$, and $\boldsymbol{S} = (S_1, S_2, S_3) = \left(x^4 + x^2 + 1,\ x^4 + 1,\ x^4 + x^3 + 1\right)$, with degree bounds $\boldsymbol{N} = (5, 3, 4, 5)$. Then $\lambda_1 = x^4 + 1$ is a solution, since $\deg \lambda_1 < 5$ and*

$$\lambda_1 \boldsymbol{S} \equiv \left(x^2 + 1,\ 1,\ x^3 + 1\right) \mod x^5 .$$

$\lambda_2 = x^3 + x$ is another solution, since

$$\lambda_2 \boldsymbol{S} \equiv \left(x,\ x^3 + x,\ x^4 + x^3 + x\right) \mod x^5 .$$

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

Several previous algorithms for solving Problem 1 are more ambitious and produce an entire *basis* of solutions that satisfy the first output condition $\lambda S_i \equiv \phi_i \mod g_i$ for $i = 1, \ldots, n$, including solutions that do not satisfy the degree bounds stipulated by the second output condition. Our algorithms are slightly more restricted in that we only return the subbasis that generates the set of solutions that satisfy both output requirements of Problem 1. Formally:

PROBLEM 3. *Given an instance of Problem 1, find a matrix $A \in \mathsf{K}[x]^{* \times (n+1)}$ such that:*

- *Each row of A is a solution to the instance.*

ISSAC '16, July 19 - 22, 2016, Waterloo, ON, Canada

ISBN 978-1-4503-4380-0/16/07…$15.00

DOI: http://dx.doi.org/10.1145/2930889.2930933

- *All solutions are in the* $\mathsf{K}[x]$*-row space of* A*.*
- A *is* $(-\boldsymbol{N})$*-row reduced*[1]*.*

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*. In the complexities we report here, we cannot afford to compute A explicitly. For example, if all $g_i = x^d$, the number of field elements required to explicitly write down all of the entries of A could be $\Omega(n^2 d)$. Instead, we remark that A is completely given by the problem instance as well as the first column of A, containing the λ polynomials.[2] Our algorithms will therefore represent A row-wise using the following compact representation.

DEFINITION 4. *For a given instance of Problem 3, a* solution specification *is a tuple* $(\boldsymbol{\lambda}, \boldsymbol{\delta}) \in \mathsf{K}[x]^{k\times 1} \times \mathbb{Z}^k_{<0}$ *such that the* completion *of* $\boldsymbol{\lambda}$ *is a solution basis, and where* $\boldsymbol{\delta}$ *are the* $(-\boldsymbol{N})$*-degrees of the rows of* A*.*

The completion *of* $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)^\top$ *is the matrix*

$$\begin{bmatrix} \lambda_1 & \operatorname{rem}(\lambda_1 S_1,\ g_1) & \cdots & \operatorname{rem}(\lambda_1 S_n,\ g_n) \\ \vdots & & \ddots & \vdots \\ \lambda_k & \operatorname{rem}(\lambda_k S_1,\ g_1) & \cdots & \operatorname{rem}(\lambda_k S_n,\ g_n) \end{bmatrix} .$$

Note that $\boldsymbol{\delta}$ will consist of only negative numbers, since any solution $\boldsymbol{v}$ by definition has $\deg_{-\boldsymbol{N}} \boldsymbol{v} < 0$.

EXAMPLE 5. *A solution specification for the problem in Example 2 is*

$$(\boldsymbol{\lambda}, \boldsymbol{\delta}) = \left([x^4 + 1,\ x^3 + x]^\top,\ (-1, -1)\right) .$$

The completion of this is

$$A = \begin{bmatrix} x^4 + 1 & x^2 + 1 & 1 & x^3 + 1 \\ x^3 + x & x & x^3 + x & x^4 + x^3 + x \end{bmatrix}$$

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

We present two algorithms for solving Problem 3, both with complexity $O\big(n^{\omega-1}\,\mathsf{M}(d)\,(\log d)\,(\log d/n)^2\big)$, where $d = \max_i \deg g_i$ and $\mathsf{M}(d)$ is the cost of multiplying two polynomials of degree d, see Section 1.1. They both depend crucially on recent developments that allow computing minimal approximant bases of non-square matrices faster than for the square case [19, 34]. We remark that from the solution basis, one can also compute the expanded form of one or a few of the solutions in the same complexity, for instance if a single, expanded solution to the simultaneous Padé problem is needed.

Our first algorithm in Section 4 assumes $g_i = x^d$ for all i and some $d \in \mathbb{Z}_{\geq 0}$. It utilises a well-known duality between Simultaneous Padé approximations and Hermite Padé approximations, see e.g. [3]. The Hermite Padé problem is immediately solvable by fast minimal approximant basis computation. A remaining step is to efficiently compute a single row of the adjoint of a matrix in Popov form, and this is done by combining partial linearisation [16] and high-order lifting [31].

Our second algorithm in Section 5 supports arbitrary g_i. The algorithm first solves n single-sequence Padé approximations, each of $S_1, \ldots, S_n$. The solution bases for two problem instances can be combined by computing the intersection of their row spaces; this is handled by a minimal approximant basis computation. A solution basis of the full Simultaneous Padé problem is then obtained by structuring intersections along a binary tree.

Before we describe our algorithms, we give some preliminary notation and definitions in Section 2, and in Section 3 we describe some of the computational tools that we employ.

Both our algorithms have been implemented in Sage v. 7.0 [30] (though asymptotically slower alternatives to the computational tools are used). The source code can be downloaded from http://jsrn.dk/code-for-articles.

1.1 Cost model

We count basic arithmetic operations in K on an algebraic RAM. We will state complexity results in terms of an exponent ω for matrix multiplication, and a function $\mathsf{M}(\cdot)$ that is a multiplication time for $\mathsf{K}[x]$ [33, Definition 8.26]. Then two $n \times n$ matrices over K can be multiplied in $O(n^\omega)$ operations in K, and two polynomials in $\mathsf{K}[x]$ of degree strictly less than d can be multiplied in $\mathsf{M}(d)$ operations in K. The best known algorithms allow $\omega < 2.38$ [12, 14], and we can always take $\mathsf{M}(d) \in O(n(\log n)(\log\log n))$ [9]. In

this paper we assume that $\omega > 2$, and that $\mathsf{M}(d)$ is super-linear while $\mathsf{M}(d) \in o(d^{\omega-1})$. The assumption $\mathsf{M}(d) \in o(d^{\omega-1})$ simply stipulates that if fast matrix multiplication techiques are used then fast polynomial multiplication should be used also: for example, $n\,\mathsf{M}(nd) \in O(n^\omega\,\mathsf{M}(d))$.

2. PRELIMINARIES

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

2.1 Degrees and shifted degrees

The degree of a nonzero 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. If A has no zero rows the *row degrees* of A, denoted by $\operatorname{rowdeg} A$, is the tuple $(d_1, \ldots, d_n)$ with $d_i = \deg \operatorname{row}(A, i)$.

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

Next we recall [2, 19, 34] the shifted variants of the notion of degree, row degrees, and leading matrix. For a *shift* $\boldsymbol{s} = (s_1, \ldots, s_n) \in \mathbb{Z}^n$, define the $n \times n$ diagonal matrix $x^{\boldsymbol{s}}$ by

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

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 $\deg_{\boldsymbol{s}} v := \deg v x^{\boldsymbol{s}}$,

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

[2] The restriction $N_i \leq \deg g_i$ in Problem 1 ensures that for a given λ, the only possibilities for the ϕ_i in a solution are $\operatorname{rem}(\lambda S_i,\ g_i)$. In particular, if we allowed $N_i > \deg g_i$ then $(0, \ldots, 0, g_i, 0, \ldots, 0)$ would be a solution which can not be directly reconstructed from its first element.

$\text{rowdeg}_{\boldsymbol{s}} A := \text{rowdeg}\, A\boldsymbol{x}^{\boldsymbol{s}}$, and $\text{LM}_{\boldsymbol{s}}(A) := \text{LM}(A\boldsymbol{x}^{\boldsymbol{s}})$. Note that we pass over the ring of Laurent polynomials only for convenience; our algorithms will only compute with polynomials. As pointed out in [19], up to negation the definition of $\boldsymbol{s}$-degree is equivalent to that used in [7] and to the notion of *defect* in [4].

For an instance $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$ of Problem 1, in the context of defining matrices, we will be using $\boldsymbol{S}$ and $\boldsymbol{g}$ as vectors, and by $\Gamma_{\boldsymbol{g}}$ denote the diagonal matrix with the entries of $\boldsymbol{g}$ on its diagonal.

2.2 Row 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 $R \in \mathsf{K}[x]^{n\times m}$ is *row reduced* if $\text{LM}(R)$ has full row rank, and *$\boldsymbol{s}$-row reduced* if $\text{LM}_{\boldsymbol{s}}(R)$ has full row rank. 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 rows of R give a basis for $\text{Row}(A)$ that is minimal in the following sense: the list of $\boldsymbol{s}$-degrees of the rows of R, when sorted in non-decreasing order, will be lexicographically minimal. An important feature of row reduced matrices is the so-called "predictable degree"-property [21, Theorem 6.3-13]: for any $\boldsymbol{v} \in \mathsf{K}[x]^{1\times n}$, we have

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

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

DEFINITION 6. *A non-singular matrix $R \in \mathsf{K}[x]^{n\times n}$ is in $\boldsymbol{s}$-Popov form if $\text{LM}_{\boldsymbol{s}}(R)$ 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.*

2.3 Adjoints of row reduced matrices

For a non-singular matrix A recall that the adjoint of A, denoted by $\text{adj}(A)$, is equal to $(\det A)A^{-1}$, and that entry $\text{adj}(A)^{\top}_{i,j}$ is equal to $(-1)^{i+j}$ times the determinant of the $(n-1)\times(n-1)$ sub-matrix that is obtained from A by deleting row i and column j.

LEMMA 7. *Let $A \in \mathsf{K}[x]^{n\times n}$ be $\boldsymbol{s}$-row reduced. Then $\text{adj}(A)^{\top}$ is $(-\boldsymbol{s})$-row reduced with*

$$\text{rowdeg}_{(-\boldsymbol{s})}\text{adj}(A)^{\top} = (\eta - s - \eta_1, \ldots, \eta - s - \eta_n) \ ,$$

where $\boldsymbol{\eta} = \text{rowdeg}_{\boldsymbol{s}} A$, $\eta = \sum_i \eta_i$ and $s = \sum_i s_i$.

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

2.4 Minimal approximant bases

We recall the standard notion of minimal approximant basis, sometimes known as order basis or σ-basis [4]. For a matrix $A \in \mathsf{K}[x]^{n\times m}$ and order $d \in \mathbb{Z}_{\geq 0}$, an *order d approximant* is a vector $\boldsymbol{p} \in \mathsf{K}[x]^{1\times n}$ such that $\boldsymbol{p}A \equiv \boldsymbol{0} \bmod x^d$.

An *approximant basis of order d* is then a matrix $F \in \mathsf{K}[x]^{n\times n}$ which is a basis of all order d approximants. Such a basis always exists and has full rank n. For a shift $\boldsymbol{s} \in \mathbb{Z}^n$, F is then an *$\boldsymbol{s}$-minimal approximant basis* if it is $\boldsymbol{s}$-row reduced.

Let $\mathsf{MinBasis}(d, A, \boldsymbol{s})$ be a function that returns $(F, \boldsymbol{\delta})$, where F is an $\boldsymbol{s}$-minimal approximant basis of A of order d, and $\boldsymbol{\delta} = \text{rowdeg}_{\boldsymbol{s}} F$. The next lemma recalls a well known method of constructing minimal approximant bases recursively. Although the output of $\mathsf{MinBasis}$ may not be unique, the lemma holds for *any* $\boldsymbol{s}$-minimal approximant basis that $\mathsf{MinBasis}$ might return.

LEMMA 8. *Let $A = \left[\begin{array}{c|c} A_1 & A_2 \end{array}\right]$ over $\mathsf{K}[x]$. If $(F_1, \boldsymbol{\delta}_1) = \mathsf{MinBasis}(d, A_1, \boldsymbol{s})$ and $(F_2, \boldsymbol{\delta}_2) = \mathsf{MinBasis}(d, F_1A_2, \boldsymbol{\delta}_1)$, then F_2F_1 is an $\boldsymbol{s}$-minimal approximant basis of A of order d with $\boldsymbol{\delta}_2 = \text{rowdeg}_{\boldsymbol{s}} F_2F_1$.*

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

COROLLARY 9. *Lemma 8 still holds if $\mathsf{MinBasis}$ is replaced by $\mathsf{NegMinBasis}$, and "an $\boldsymbol{s}$-minimal" is replaced with "the negative part of an $\boldsymbol{s}$-minimal."*

Using for example the algorithm `M-Basis` of [15], it is easy to show that any order d approximant basis G for an A of column dimension m has $\det G = x^D$ for some $D \in \mathbb{Z}_{\geq 0}$ with $D \leq md$.

Many problems of $\mathsf{K}[x]$ matrices or approximations reduce to the computation of (shifted) minimal approximant bases, see e.g. [4, 15], often resulting in the best known asymptotic complexities for these problems.

2.5 Direct solving of Simultaneous Padé approximations

Let $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$ be an instance of Problem 3 of size n. We recall some known approaches for computing a solution specification using row reduction and minimal approximant basis computation.

2.5.1 Via reduced basis

Using the predictable degree property it is easy to show that if $R \in \mathsf{K}[x]^{(n+1)\times(n+1)}$ is an $(-\boldsymbol{N})$-reduced basis of

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

then the sub-matrix of R comprised of the rows with negative $(-\boldsymbol{N})$-degree form a solution basis. A solution specification $(\boldsymbol{\lambda}, \boldsymbol{\delta})$ is then a subvector $\boldsymbol{\lambda}$ of the first column of R, with $\boldsymbol{\delta}$ the corresponding subtuple $\boldsymbol{\delta}$ of $\text{rowdeg}_{(-\boldsymbol{N})} R$.

Mulders and Storjohann [24] gave an iterative algorithm for performing row reduction by successive cancellation; it is similar to but faster than earlier algorithms [21, 22]. Generically on input $F \in \mathsf{K}[x]^{m\times m}$ it has complexity $O(n^3(\deg F)^2)$.

Alekhnovich [1] gave what is essentially a Divide & Conquer variant of Mulders and Storjohann's algorithm, with complexity $O^{\sim}(n^{\omega+1}\deg F)$. Nielsen remarked [26] that these algorithms perform fewer iterations when applied to the matrix A above, due to its low *orthogonality defect*: $\mathrm{OD}(F) = \sum \mathrm{rowdeg} F - \deg\det F$, resulting in $O(n^2(\deg A)^2)$ respectively $O^{\sim}(n^{\omega}\deg A)$. Nielsen also used the special shape of A to give a variant of the Mulders–Storjohann algorithm that computes coefficients in the working matrix in a lazy manner with a resulting complexity $O(n\,\mathsf{P}(\deg A))$, where $\mathsf{P}(\deg A) = (\deg A)^2$ when the g_i are all powers of x, and $\mathsf{P}(\deg A) = \mathsf{M}(\deg A)\deg A$ otherwise.

Giorgi, et al. [15] gave a reduction for performing row reduction by computing a minimal approximant basis. For the special matrix A, this essentially boils down to the approach described in the following section.

When $n = 1$, the extended Euclidean algorithm on input S_1 and g_1 can solve the approximation problem by essentially computing the reduced basis of the 2×2 matrix A: each iteration corresponds to a reduced basis for a range of possible shifts [17, 20, 32]. The complexity of this is $O(\mathsf{M}(\deg g_1)\log\deg g_1)$.

2.5.2 Via minimal approximant basis

First consider the special case when all $g_i = x^d$ for the same d. An approximant $\boldsymbol{v} = (\lambda, \phi_1, \ldots, \phi_n)$ of order d of

$$A = \begin{bmatrix} -\boldsymbol{S} \\ I \end{bmatrix} \in \mathsf{K}[x]^{(n+1)\times n}$$

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

In the general case we can reduce to a minimal approximant bases computation as shown by Algorithm 1. Correctness of the algorithm follows from the following result.

THEOREM 10. *Corresponding to an instance $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$ of Problem 3 of size n, define a shift $\boldsymbol{h}$ and order d:*

- $\boldsymbol{h} := -(\boldsymbol{N} \mid N_0 - 1, \ldots, N_0 - 1) \in \mathbb{Z}^{2n+1}$
- $d := N_0 + \max_i \deg g_i - 1$

If G is the negative part of an $\boldsymbol{h}$-minimal approximant basis of

$$H = \begin{bmatrix} -\boldsymbol{S} \\ I \\ \Gamma_{\boldsymbol{g}} \end{bmatrix} \in \mathsf{K}[x]^{(2n+1)\times n}$$

of order d, then the submatrix of G comprised of the first $n+1$ columns is a solution basis to the problem instance.

PROOF. An approximant $\boldsymbol{v} = (\lambda, \phi_1, \ldots, \phi_n, q_1, \ldots, q_n)$ of order d of H clearly satisfies

$$\lambda S_i = \phi_i + q_i g_i \bmod x^d \tag{1}$$

for $i = 1, \ldots, n$; conversely, any such vector $\boldsymbol{v}$ satisfying these congruences must be an approximant of H of order d.

Now suppose $\boldsymbol{v}$ is an order d approximant of H with negative $\boldsymbol{h}$-degree, so $\deg\lambda \le N_0 - 1$, $\deg\phi_i \le N_i - 1$, and $\deg q_i \le N_0 - 2$. Since Problem 1 specifies that $\deg S_i < \deg g_i$ and $N_i \le \deg g_i$, both λS_i and $q_i g_i$ will have degree bounded by $N_0 + \deg g_i - 2$. Since Problem 1 specifies that $N_0 \ge 1$, it follows that both the left and right hand sides of (1) have degree bounded by $N_0 + \deg g_i - 2$, which is strictly less than d. We conclude that

$$\lambda S_i = \phi_i + q_i g_i \tag{2}$$

for $i = 1, \ldots, n$. It follows that $\boldsymbol{v}H = 0$ so $\boldsymbol{v}$ is in the left kernel of H. Moreover, restricting $\boldsymbol{v}$ to its first $n+1$ entries gives $\bar{\boldsymbol{v}} := (\lambda, \phi_1, \ldots, \phi_n)$, a solution to the simultaneous Padé problem with $\deg_{-\boldsymbol{N}} \bar{\boldsymbol{v}} = \deg_{\boldsymbol{h}} \boldsymbol{v}$. Conversely, if $\bar{\boldsymbol{v}} = (\lambda, \phi_1, \ldots, \phi_n)$ is a solution to the simultaneous Padé problem, then the extension $\boldsymbol{v} = (\lambda, \phi_1, \ldots, \phi_n, q_1, \ldots, q_n)$ with $q_i = (\lambda S_i - \phi_i)/g_i \in \mathsf{K}[x]$ for $i = 1, \ldots, n$ is an approximant of H of order d with $\deg_{\boldsymbol{h}} \boldsymbol{v} = \deg_{-\boldsymbol{N}} \bar{\boldsymbol{v}}$.

Finally, consider that a left kernel basis for H is given by

$$K = \begin{bmatrix} K_1 \mid K_2 \end{bmatrix} = \left[\begin{array}{c|c} 1 & \boldsymbol{S} \\ & \Gamma_{\boldsymbol{g}} \end{array} \middle| \begin{array}{c} \\ -I \end{array}\right].$$

We must have $G = MK$ for some polynomial matrix M of full row rank. But then MK_1 also has full row rank with $\mathrm{rowdeg}_{-\boldsymbol{N}} MK_1 = \mathrm{rowdeg}_{\boldsymbol{h}} G$. □

Algorithm 1 DirectSimPade

Input: $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$, an instance of Problem 3 of size n.
Output: $(\boldsymbol{\lambda}, \boldsymbol{\delta})$, a solution specification.

```
h ← −(N | N_0 − 1, ..., N_0 − 1) ∈ Z^{2n+1}
d ← N_0 + max_i deg g_i − 1
H = [ −S ]
    [  I ]
    [ Γ_g ]
([ λ | * ], δ) ← NegMinBasis(d, H, h)
return (λ, δ)
```

DirectSimPade can be performed in time $O^{\sim}(n^{\omega}\deg H) = O^{\sim}(n^{\omega}\max_i \deg g_i)$ using the minimal approximant basis algorithm by Jeannerod, et al. [19], see Section 3.

A closely related alternative to DirectSimPade is the recent algorithm by Neiger [25] for computing solutions to modular equations with general moduli g_i. This would give the complexity $O^{\sim}(n^{\omega-1}\sum_i \deg g_i) \subset O^{\sim}(n^{\omega}\max_i \deg g_i)$.

All of the above solutions ignore the sparse, simple structure of the input matrices, which is why they do not obtain the improved complexity that we do here.

3. COMPUTATIONAL TOOLS

The main computational tool we will use is the following very recent result from Jeannerod, Neiger, Schost and Villard [19] on minimal approximant basis computation.

THEOREM 11 ([19, SPECIAL CASE OF THEOREM 1.4]). *There exists an algorithm* PopovBasis$(d, A, \boldsymbol{s})$ *where the input is an order $d \in \mathbb{Z}_+$, a polynomial matrix $A \in \mathsf{K}[x]^{n\times m}$ of degree at most d, and shift $\boldsymbol{s} \in \mathbb{Z}^n$, and which returns $(F, \boldsymbol{\delta})$, where F is an $\boldsymbol{s}$-minimal approximant basis of A of order d, F is in $\boldsymbol{s}$-Popov form, and $\boldsymbol{\delta} = \mathrm{rowdeg}_{\boldsymbol{s}} F$.* PopovBasis *has complexity $O(n^{\omega-1}\,\mathsf{M}(\sigma)\,(\log\sigma)\,(\log\sigma/n)^2)$ operations in* K, *where $\sigma = md$.*

Our next result says that we can quickly compute the first row of $\mathrm{adj}(F)$ if F is a minimal approximant basis in Popov form. In particular, since F is an approximant basis $\det F = x^D$ for some $D \le \sigma$, where $\sigma = md$ from Theorem 11.

THEOREM 12. *Let $F \in \mathsf{K}[x]^{n\times n}$ be in Popov form and with $\det F = x^D$ for some $D \in \mathbb{Z}_{\geq 0}$. Then the first row of $\mathrm{adj}(F)$ can be computed in $O(n^{\omega-1}\,\mathsf{M}(D)\,(\log D)\,(\log D/n))$ operations in K.*

PROOF. Because F is in $\boldsymbol{s}$-Popov form, D is the sum of the column degrees of F. We consider two cases: $D \geq n$ and $D < n$.

First suppose $D \geq n$. Partial linearisation [16, Corollary 2] can produce from F, with no operations in K, a new matrix $G \in \mathsf{K}[x]^{\bar{n}\times\bar{n}}$ with dimension $\bar{n} < 2n$, $\deg G \leq \lceil D/n \rceil$, $\det G = \det F$, and such that F^{-1} is equal to the principal $n \times n$ sub-matrix of G^{-1}. Let $\boldsymbol{v} \in \mathsf{K}[x]^{1\times\bar{n}}$ be the first row of $x^D I_{\bar{n}}$. Then the first row of $\mathrm{adj}(F)$ will be the first n entries of the first row of $\boldsymbol{v}G^{-1}$. High-order X-adic lifting [31, Algorithm 5] using the modulus $X = (x-1)^{\lceil D/n \rceil}$ will compute $\boldsymbol{v}G^{-1}$ in $O\big(n^\omega\,\mathsf{M}(\lceil D/n \rceil)\,(\log \lceil D/n \rceil)\big)$ operations in K [31, Corollary 16]. Since $D \geq n$ this cost estimate remains valid if we replace $\lceil D/n \rceil$ with D/n. Finally, from the superlinearity assumption on $\mathsf{M}(\cdot)$ we have $M(D/n) \leq (1/n)\mathsf{M}(D)$, thus matching our target cost.

Now suppose $D < n$. In this case we can not directly appeal to the partial linearisation technique since the resulting $O(n^\omega \lceil D/n \rceil)$ may be asymptotically larger than our target cost. But $D < n$ means that F has — possibly many — columns of degree 0; 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 those columns. D is then greater than or equal to the number of remaining columns, thus effectuating the gain from the partial linearisation.

If $n - k$ is the number of such columns in F that means we can find a permutation matrix P such that

$$\hat{F} := PFP^\top = \left[\begin{array}{c|c} F_1 & \\ \hline F_2 & I_{n-k} \end{array}\right] ,$$

with each column of F_1 having degree strictly greater than zero. Let i be the row index of the single 1 in the first column of $P^\top$. Since $F^{-1} = P^\top \hat{F}^{-1} P$, we have

$$\mathrm{row}(\mathrm{adj}(F), 1)P^{-1} = x^D\,\mathrm{row}(\hat{F}^{-1}, i). \tag{3}$$

Considering that

$$\hat{F}^{-1} = \left[\begin{array}{c|c} F_1^{-1} & \\ \hline -F_2F_1^{-1} & I_{n-k} \end{array}\right],$$

it will suffice to compute the first k entries of the vector on the right hand side of (3). If $i \leq k$ then let $\boldsymbol{v} \in \mathsf{K}[x]^{1\times k}$ be row i of $x^D I_k$. Otherwise, if $i > k$ then let $\boldsymbol{v}$ be row $i-k$ of $-x^D F_2$. Then in both cases, $\boldsymbol{v}F_1^{-1}$ will be equal to the first k entries of the vector on the right hand side of (3). Like before, high-order lifting combined with partial linearisation will compute this vector in $O\big(k^\omega\,\mathsf{M}(\lceil D/k \rceil)\,(\log \lceil D/k \rceil)\big)$ operations in K. Since $D \geq k$ the cost estimate remains valid if $\lceil D/k \rceil$ is replaced with D/k. □

4. REDUCTION TO HERMITE PADÉ

In this section we present an algorithm for solving Problem 3 when $g_1 = \ldots = g_n = x^d$ for some $d \in \mathbb{Z}_{\geq 0}$. The algorithm is based on the well-known duality between the Simultaneous Padé problem and the Hermite Padé problem, see for example [3]. This duality, first observed in a special case [23], and then later in the general case [5], was exploited in [6] to develop algorithms for the fraction free computation of Simultaneous Padé approximation. We begin with a technical lemma that is at the heart of this duality.

LEMMA 13. *Let $\hat{A}, \hat{B} \in \mathsf{K}[x]^{(n+1)\times(n+1)}$ be as follows.*

$$\hat{A} = \left[\begin{array}{c|c} x^d & -\boldsymbol{S} \\ \hline & I \end{array}\right] \qquad \hat{B} = \left[\begin{array}{c|c} 1 & \\ \hline \boldsymbol{S}^\top & x^d I \end{array}\right]$$

Then $\hat{B}$ is the adjoint of $\hat{A}^\top$. Furthermore, $\hat{A}^\top$ is an approximant basis for $\hat{B}$ of order d, and $\hat{B}^\top$ is an approximant basis of $\hat{A}$ of order d.

PROOF. Direct computation shows that $\hat{A}^\top\hat{B} = x^d I_m = \det \hat{A}^\top I_m$, so $\hat{B}$ is the adjoint of $\hat{A}^\top$.

Let now G be an approximant basis of $\hat{B}$. By the above computation the row space of $\hat{A}^\top$ must be a subset of the row space of G. But since $G\hat{B} = (x^d I_m)R$ for some $R \in \mathsf{K}[x]^{(n+1)\times(n+1)}$, then $\det G = x^d \det R$. Thus $x^d \mid \det G$. But $\det \hat{A}^\top = x^d$, so the row space of $\hat{A}^\top$ can not be smaller than the row space of G. That is, $\hat{A}^\top$ is an approximant basis for B of order d. Taking the transpose through the argument shows that $\hat{B}^\top$ is an approximant basis of $\hat{B}$ of order d. □

THEOREM 14. *Let A and B be as follows.*

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

If G is an $\boldsymbol{N}$-minimal approximant basis of B of order d with shift $\boldsymbol{N} \in \mathbb{Z}_{\geq 0}^{n+1}$, then $\mathrm{adj}(G^\top)$ is a $(-\boldsymbol{N})$-minimal approximant basis of A of order d. Moreover, if $\boldsymbol{\eta} = \mathrm{rowdeg}_{\boldsymbol{N}} G$, then $\mathrm{rowdeg}_{-\boldsymbol{N}}\mathrm{adj}(G) = (\eta - N - \eta_1, \ldots, \eta - N - \eta_{n+1})$, where $\eta = \sum_i \eta_i$ and $N = \sum_i N_i$.

PROOF. Introduce $\hat{A}$ and $\hat{B}$ as in Lemma 13. Clearly G is also an $\boldsymbol{N}$-minimal approximant basis of $\hat{B}$ of order d. Likewise, $\hat{A}$ and A have the same minimal approximant bases for given order and shift.

Assume, without loss of generality, that we have scaled G such that $\det G$ is monic. Since $\hat{A}^\top$ is also an approximant basis for $\hat{B}$ of order d, then $\det G = \det \hat{A}^\top = x^d$. By definition $G\hat{B} = x^d R$ for some matrix $R \in \mathsf{K}[x]^{(n+1)\times(n+1)}$. That means

$$\begin{aligned} x^{2d}((G\hat{B})^\top))^{-1} &= x^{2d}((x^d R)^\top)^{-1} , && \text{so} \\ (x^d(G^\top)^{-1})(x^d(\hat{B}^\top)^{-1}) &= x^d(R^\top)^{-1} , && \text{that is} \\ \mathrm{adj}(G^\top)\hat{A} &= x^d(R^\top)^{-1} . \end{aligned}$$

Now $\det R = 1$ since $(x^d)^{n+1} \det R = \det(G\hat{B}) = x^{d+nd}$, so $(R^\top)^{-1} = \mathrm{adj}(R^\top) \in \mathsf{K}[x]^{(n+1)\times(n+1)}$. Therefore $\mathrm{adj}(G^\top)$ is an approximant basis of $\hat{A}$ of order d. The theorem now follows from Lemma 7 by noting that G is $\boldsymbol{N}$-row reduced. □

EXAMPLE 15. *We apply Theorem 14 to the problem of Example 2 with shifts $\boldsymbol{N} = (5, 3, 4, 5)$. We have*

$$A = \begin{bmatrix} x^4+x^2+1 & x^4+1 & x^4+x^3+1 & \\ 1 & & & \\ & 1 & & \\ & & & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ x^4+x^2+1 \\ x^4+1 \\ x^4+x^3+1 \end{bmatrix}$$

Algorithm 2 DualitySimPade

Input: $(\boldsymbol{S}, (x^d, \ldots, x^d), \boldsymbol{N})$, an instance of Problem 3 of size n.

Output: $(\boldsymbol{\lambda}, \boldsymbol{\delta})$, solution specification.

1 $B \leftarrow [1, S_1, \ldots, S_n]^T \in \mathsf{K}[x]^{(n+1)\times 1}$
2 $G \leftarrow \mathsf{PopovBasis}(d, B, \boldsymbol{N})$
3 $\boldsymbol{\eta} \leftarrow \mathrm{rowdeg}_{\boldsymbol{N}} G$
4 $\hat{\boldsymbol{\lambda}} \leftarrow$ first column of $\mathrm{adj}(G^\top)$
5 $\hat{\boldsymbol{\delta}} \leftarrow (\eta - N - \eta_1, \ldots, \eta - N - \eta_{n+1})$, where $\eta = \sum_i \eta_i$ and $N = \sum_i N_i$
6 $I \leftarrow \{i \mid \hat{\boldsymbol{\delta}}_i < 0\}$, and $k \leftarrow |I|$
7 $(\boldsymbol{\lambda},\ \boldsymbol{\delta}) \leftarrow \left(\hat{\boldsymbol{\lambda}}_{i\in I},\ (\hat{\boldsymbol{\delta}}_i)_{i\in I}\right) \in \mathsf{K}[x]^{k\times 1} \times \mathbb{Z}^k$
8 **return** $(\boldsymbol{\lambda}, \boldsymbol{\delta})$

An $\boldsymbol{N}$-minimal approximant basis to order $d = 5$ of B is

$$G = \begin{bmatrix} x & 0 & x & 0 \\ 1 & x^2+1 & 0 & 0 \\ 0 & 1 & x^2+1 & 0 \\ 0 & x & x+1 & 1 \end{bmatrix}, \text{ and}$$

$$\mathrm{adj}(G)^\top = \begin{bmatrix} x^4+1 & x^2+1 & 1 & x^3+1 \\ x & x^3+x & x & x^4+x \\ x^3+x & x & x^3+x & x^4+x^3+x \\ 0 & 0 & 0 & x^5 \end{bmatrix}.$$

$\mathrm{adj}(G)^\top$ can be confirmed to be an $(-\boldsymbol{N})$-minimal approximant basis of A, since $\mathrm{adj}(G)^\top A \equiv 0 \mod x^d$, and since the $(-\boldsymbol{N})$-leading coefficient matrix of $\mathrm{adj}(G)^\top$ has full rank.

Algorithm 2 uses Theorem 14 to solve a Simultaneous Padé approximation by computing a minimal approximant basis of B in Popov form.

THEOREM 16. *Algorithm 2 is correct. The cost of the algorithm is $O(n^{\omega-1}\,\mathsf{M}(d)(\log d)(\log d/n)^2)$ operations in K.*

PROOF. Correctness follows from Theorem 14. The complexity estimate is achieved if the algorithms supporting Theorem 11 and Theorem 12 are used for the computation in lines 2 and 4, respectively. □

5. A DIVIDE & CONQUER ALGORITHM

Our second algorithm can handle the full generality of Problem 3. It works by first solving n single Padé approximations, one for each of the S_i individually, and then intersecting these solutions to form approximations of multiple S_i simultaneously. The intersection is structured in a Divide & Conquer tree, and performed by computing minimal approximant bases. Let $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$ be an instance of Problem 3 of size n.

The idea of the intersection algorithm is the following: consider that we have solution specifications for two different Simultaneous Padé problems, $(\boldsymbol{\lambda}_1, \boldsymbol{\delta}_1)$ and $(\boldsymbol{\lambda}_2, \boldsymbol{\delta}_2)$. We then compute an approximant basis G of the following matrix:

$$R = \left[\begin{array}{c|c} 1 & 1 \\ \hline -\boldsymbol{\lambda}_1 & \\ \hline & -\boldsymbol{\lambda}_2 \end{array}\right] \tag{4}$$

G then encodes the *intersection* of the $\mathsf{K}[x]$-linear combinations of the $\boldsymbol{\lambda}_1$ with the $\mathsf{K}[x]$-linear combinations of the $\boldsymbol{\lambda}_2$: any $\lambda \in \mathsf{K}[x]$ residing in both sets of polynomials will appear as the first entry of a vector in the row space of G. We compute G as an $\boldsymbol{r}$-minimal approximant basis to high enough order, where $\boldsymbol{r}$ is selected carefully such that the $\boldsymbol{r}$-degree of any $(\lambda \mid \ldots) \in \mathrm{Row}(G)$ will equal the $(-\boldsymbol{N})$-degree of the completion of λ according to the combined Simultaneous Padé problem, whenever this degree is negative. From those rows of G with negative $\boldsymbol{r}$-degree we then get a solution specification for the combined problem.

EXAMPLE 17. *Consider again Example 2. We divide the problem into two sub-problems $\boldsymbol{S}_1 = (S_1, S_2)$, $\boldsymbol{N}_1 = (5, 3, 4)$, and $\boldsymbol{S}_2 = (S_3)$ and $\boldsymbol{N}_2 = (5, 5)$. Note that $N_{1,0} = N_{2,0} = 5$, 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:*

$$(\boldsymbol{\lambda}_1, \boldsymbol{\delta}_1) = \left([x^4+1,\ x^3+x]^\top,\ (-1,-1)\right)$$
$$A_1 = \begin{pmatrix} x^4+1 & x^2+1 & 1 \\ x^3+x & x & x^3+x \end{pmatrix}$$
$$(\boldsymbol{\lambda}_2, \boldsymbol{\delta}_2) = \left([x^2,\ x^3+x+1]^\top,\ (-3,-2)\right)$$
$$A_2 = \begin{pmatrix} x^2 & x^2 \\ x^3+x+1 & x+1 \end{pmatrix}$$

We construct R as in (4), and compute G, a minimal approximant basis of R of order 7 and with shifts $\boldsymbol{r} = (-5 \mid -1, -1 \mid -3, -2)$ (the G below is actually in $\boldsymbol{r}$-Popov form):

$$G = \begin{pmatrix} x^8 & 0 & 0 & 0 & 0 \\ x^3+x+1 & x^4+1 & 1 & 0 & 1 \\ x^3+x^2+x+1 & 1 & x+1 & 1 & 1 \\ x^4+x^3+x+1 & 1 & 1 & x^2 & 1 \\ x^4+1 & 1 & 0 & x+1 & x+1 \end{pmatrix}$$

G has $\boldsymbol{r}$-row degree $(3, 3, 0, -1, -1)$. Only rows 4 and 5 have negative $\boldsymbol{r}$-degree, and their first entries are the linearly independent solutions x^4+x^3+x+1 and x^4+1. Both solutions complete into vectors with $(-\boldsymbol{N})$-degree -1.

To prove the correctness of the above intuition, we will use Algorithm 1 (DirectSimPade). The following lemma says that to solve two simultaneous Padé approximations, one can compute a minimal approximant basis of one big matrix A constructed essentially from two of the matrices employed in DirectSimPade. Afterwards, Lemma 19 uses this to show that a minimal approximant basis of R in (4) provides the crucial information in a minimal approximant basis of A.

LEMMA 18. *Let $(\boldsymbol{S}_1, \boldsymbol{g}_1, \boldsymbol{N}_1)$ and $(\boldsymbol{S}_2, \boldsymbol{g}_2, \boldsymbol{N}_2)$ be two instances of Problem 3 of lengths n_1, n_2 respectively, and where $\boldsymbol{N}_1 = (N_0 \mid \hat{\boldsymbol{N}}_1)$ and $\boldsymbol{N}_2 = (N_0 \mid \hat{\boldsymbol{N}}_2)$. Let $\boldsymbol{S} = (\boldsymbol{S}_1 \mid \boldsymbol{S}_2)$, $\boldsymbol{g} = (\boldsymbol{g}_1 \mid \boldsymbol{g}_2)$ and $\boldsymbol{N} = (N_0 \mid \hat{\boldsymbol{N}}_1 \mid \hat{\boldsymbol{N}}_2)$ be the combined problem having length $n = n_1 + n_2$.*

Let $\boldsymbol{h}_i = (-\boldsymbol{N}_i \mid N_0 - 1 \ldots N_0 - 1) \in \mathbb{Z}^{2n_i+1}$ for $i = 1, 2$. Let $(F, \boldsymbol{\delta}) = \mathsf{NegMinBasis}(d, A, \boldsymbol{a})$, where A of dimension $(2n+3) \times (n+2)$ is given as:

$$A = \left[\ A_1 \mid A_2\ \right] = \left[\begin{array}{cc|cc} & & 1 & 1 \\ -\boldsymbol{S}_1 & & -1 & \\ I & & & \\ \Gamma_{\boldsymbol{g}_1} & & & \\ & -\boldsymbol{S}_2 & & -1 \\ & I & & \\ & \Gamma_{\boldsymbol{g}_2} & & \end{array}\right],$$

Algorithm 3 RecursiveSimPade

Input: $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$, an instance of Problem 3 of size n.
Output: $(\boldsymbol{\lambda}, \boldsymbol{\delta})$, a solution specification.

1 **if** $n = 1$ **then**
2 **return** $\mathsf{DirectSimPade}(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$
3 **else**
4 $\boldsymbol{S}_1, \boldsymbol{g}_1 \leftarrow$ the first $\lceil n/2 \rceil$ elements of $\boldsymbol{S}, \boldsymbol{g}$
5 $\boldsymbol{S}_2, \boldsymbol{g}_2 \leftarrow$ the last $\lfloor n/2 \rfloor$ elements of $\boldsymbol{S}, \boldsymbol{g}$
6 $\boldsymbol{N}_1 \leftarrow (N_0, N_1, \ldots, N_{\lceil n/2 \rceil})$
7 $\boldsymbol{N}_2 \leftarrow (N_0, N_{\lceil n/2 \rceil + 1}, \ldots, N_n)$
8 $(\boldsymbol{\lambda}_1, \boldsymbol{\delta}_1) \leftarrow \mathsf{RecursiveSimPade}(\boldsymbol{S}_1, \boldsymbol{g}_1, \boldsymbol{N}_1)$
9 $(\boldsymbol{\lambda}_2, \boldsymbol{\delta}_2) \leftarrow \mathsf{RecursiveSimPade}(\boldsymbol{S}_2, \boldsymbol{g}_2, \boldsymbol{N}_2)$
10 $\boldsymbol{r} \leftarrow (-N_0 \mid \boldsymbol{\delta}_1 \mid \boldsymbol{\delta}_2)$
11 $d \leftarrow N_0 + \max_i \deg g_i - 1$
12 $R \leftarrow \left[\begin{array}{c|c} 1 & 1 \\ \hline -\boldsymbol{\lambda}_1 & \\ \hline & -\boldsymbol{\lambda}_2 \end{array}\right]$
13 $(\left[\ \boldsymbol{\lambda} \mid * \ \right], \boldsymbol{\delta}) \leftarrow \mathsf{NegMinBasis}(d, R, \boldsymbol{r})$
14 **return** $(\boldsymbol{\lambda}, \boldsymbol{\delta})$
15 **end if**

with $\boldsymbol{a} = (-N_0 \mid \boldsymbol{h}_1 \mid \boldsymbol{h}_2)$ *and* $d = N_0 + \max_i \deg g_i - 1$. *Then* $(\boldsymbol{\lambda}, \boldsymbol{\delta})$ *is a solution specification to* $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$, *where* $\boldsymbol{\lambda}$ *is the first column of* F.

Proof. Note that the matrix A is right equivalent to the following matrix B:

$$B := A \begin{bmatrix} & I & \\ & & I \\ 1 & \boldsymbol{S}_1 & \\ & 1 & \boldsymbol{S}_2 \end{bmatrix} = \left[\begin{array}{cc|cc} 1 & 1 & -\boldsymbol{S}_1 & -\boldsymbol{S}_2 \\ -1 & & I & \\ & & \Gamma_{\boldsymbol{g}_1} & \\ & -1 & & I \\ & & & \Gamma_{\boldsymbol{g}_2} \end{array}\right].$$

Since F is an $\boldsymbol{a}$-minimal approximant of A of order d, then it will also be one for B. Let P be the permutation matrix that produces the following matrix $C := PB$:

$$C = PB = \left[\begin{array}{cc|cc} 1 & 1 & -\boldsymbol{S}_1 & -\boldsymbol{S}_2 \\ & & I & \\ & & & I \\ & & \Gamma_{\boldsymbol{g}_1} & \\ & & & \Gamma_{\boldsymbol{g}_2} \\ \hline -1 & & & \\ & -1 & & \end{array}\right] = \left[\begin{array}{cc|c} 1 & 1 & -\boldsymbol{S} \\ & & I \\ & & \Gamma_{\boldsymbol{g}} \\ \hline -1 & & \\ & -1 & \end{array}\right].$$

Define $\boldsymbol{c} := \boldsymbol{a}P^{-1}$, and note that $\boldsymbol{c} = (\boldsymbol{h} \mid -N_0, -N_0)$. Since $F = \mathsf{NegMinBasis}(d, A, \boldsymbol{a})$, then $(FP^{-1}, \boldsymbol{\delta})$ is a valid output of $\mathsf{NegMinBasis}(d, C, \boldsymbol{c})$. Furthermore, since the first column of P is $(1, 0, \ldots, 0)$, the first column of F will be equal to the first column of FP^{-1}.

We are therefore finished if we can show that if $(F', \boldsymbol{\delta}')$ is any valid output of $\mathsf{NegMinBasis}(d, C, \boldsymbol{c})$, then the first column of F' together with $\boldsymbol{\delta}'$ form a solution specification to $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$.

Consider therefore such an $(F', \boldsymbol{\delta}')$. By the first two columns of C, we must have $F'_{*,1} \equiv F'_{*,2n+2} \equiv F'_{*,2n+3} \bmod x^d$, where $F'_{*,i}$ denotes the i'th column of F'. Since each row of F' have negative $\boldsymbol{c}$-degree, and since $N_0 < d$, then the congruences must lift to equalities. We can therefore write $F = [G \mid F'_{*,1} \mid F'_{*,1}]$ for some $G \in \mathsf{K}[x]^{k \times (2n+1)}$ for some k, and we have $\operatorname{rowdeg}_{\boldsymbol{h}} G = \operatorname{rowdeg}_{\boldsymbol{c}} F' = \boldsymbol{\delta}'$.

By the last n columns of C, we have $GH \equiv 0 \mod x^d$, where

$$H = \begin{bmatrix} -\boldsymbol{S} \\ I \\ \Gamma_{\boldsymbol{g}} \end{bmatrix}.$$

In fact, $(G, \boldsymbol{\delta}')$ is a valid output for $\mathsf{NegMinBasis}(d, H, \boldsymbol{h})$: for G has full row rank since F' does; G is $\boldsymbol{h}$-row reduced since F' is $\boldsymbol{c}$-row reduced; and any negative $\boldsymbol{h}$-order d approximant of H must clearly be in the span of G since F' is a negative $\boldsymbol{c}$-minimal approximant basis of C.

By the choice of d, then Theorem 10 therefore implies that the first column of G together with $\boldsymbol{\delta}'$ form a solution specification to the problem $(\boldsymbol{S}, \boldsymbol{g}, \boldsymbol{N})$. Since the first column of G is also the first column of F', this finishes the proof. □

Lemma 19. *In the context of Lemma 18, let* $(\boldsymbol{\lambda}_1, \boldsymbol{\delta}_1)$ *and* $(\boldsymbol{\lambda}_2, \boldsymbol{\delta}_2)$ *be solution specifications to the two sub-problems, and let* $\boldsymbol{r} = (-N_0 \mid \boldsymbol{\delta}_1 \mid \boldsymbol{\delta}_2)$. *If* $([\boldsymbol{\lambda} \mid *], \boldsymbol{\delta}) = \mathsf{NegMinBasis}(d, R, \boldsymbol{r})$, *where* $\boldsymbol{\lambda}$ *is a column vector and*

$$R = \left[\begin{array}{c|c} 1 & 1 \\ \hline -\boldsymbol{\lambda}_1 & \\ \hline & -\boldsymbol{\lambda}_2 \end{array}\right],$$

then $(\boldsymbol{\lambda}, \boldsymbol{\delta})$ *is a solution specification for the combined problem.*

Proof. We will prove the lemma by using Lemma 9 to relate valid outputs of $\mathsf{NegMinBasis}(d, R, \boldsymbol{r})$ with valid outputs of $\mathsf{NegMinBasis}(d, A, \boldsymbol{a})$ from Lemma 18.

For $i = 1, 2$, since $(\boldsymbol{\lambda}_i, \boldsymbol{\delta}_i)$ is a solution specification to the i'th problem, then by Theorem 10 there is some $G_i \in \mathsf{K}[x]^{k_i \times 2n_i + 1}$ whose first column is $\boldsymbol{\lambda}_i$ and such that G_i is a valid output of $\mathsf{NegMinBasis}(d, H_i, \boldsymbol{h}_i)$, where

$$H_i = \begin{bmatrix} -\boldsymbol{S}_i \\ I \\ \Gamma_{\boldsymbol{g}_i} \end{bmatrix} \in \mathsf{K}[x]^{(2n_i+1) \times n_i},$$

and $\boldsymbol{h}_i$ is as in Lemma 18. Note now that if

$$F_1 := \begin{bmatrix} 1 & & \\ & G_1 & \\ & & G_2 \end{bmatrix} \in \mathsf{K}[x]^{(k_1+k_2+1) \times (2n_1+2n_2+3)},$$

then $(F_1, \boldsymbol{r})$ is a valid output of $\mathsf{NegMinBasis}(d, A_1, \boldsymbol{a})$: for $\operatorname{rowdeg}_{\boldsymbol{a}} F_1$ is clearly $\boldsymbol{r}$; F_1 has full row rank and is $\boldsymbol{r}$-row reduced; and the rows of F_1 must span all $\boldsymbol{a}$-order d approximants of A_1, since the three column "parts" of F_1 correspond to the three row parts of A_1. .

Note now that $F_1 A_2 = R$. Thus by Lemma 9, if $(F_2, \boldsymbol{\delta}) = \mathsf{NegMinBasis}(d, R, \boldsymbol{r})$, then $(F_2 F_1, \boldsymbol{\delta})$ is a valid output of $\mathsf{NegMinBasis}(d, A, \boldsymbol{a})$. Note that by the shape of F_1 then the first column $\boldsymbol{\lambda}$ of $F_2 F_1$ is the first column of F_2. Thus $\boldsymbol{\lambda}, \boldsymbol{\delta}$ are exactly as stated in the lemma, and by Lemma 18 they must be a solution specification to the combined problem. □

Theorem 20. *Algorithm 3 is correct. The cost of the algorithm is* $O(n^{\omega-1}\mathsf{M}(d)(\log d)(\log d/n)^2)$, $d = \max_i \deg g_i$.

Proof. Correctness follows from Lemma 19. For complexity, note that the choice of order in Line 11 is bounded by $2 \max_i \deg g_i$, i.e. twice the value of d of this theorem. So if $T(n)$ is the cost Algorithm 3 for given n and where the

order will be bounded by $O(d)$, then we have the following recursion:

$$T(n) = \begin{cases} 2T(n/2) + P(n) & \text{if } n > 1 \\ O(\mathsf{M}(d)\log d) & \text{if } n = 1 \text{ (see Section 2.5.1)} \end{cases},$$

where $P(n)$ is the cost of line 13. Using algorithm PopovBasis for the computation of the negative part of the minimal approximant bases we can set $P(n)$ to the target cost. The recursion then implies $T(n) \in O(P(n))$. □

Acknowledgements. The authors would like to thank George Labahn for valuable discussions, and for making us aware of the Hermite–Simultaneous Padé duality. We would also like to thank Vincent Neiger for making preprints of [19] available to us. The first author would like to thank the Digiteo Foundation for funding the research visit at Waterloo, during which most of the ideas of this paper were developed.

6. REFERENCES

[1] M. Alekhnovich. Linear Diophantine Equations Over Polynomials and Soft Decoding of Reed–Solomon Codes. *IEEE Trans. Inf. Th.*, 51(7):2257–2265, 2005.

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

[3] B. Beckermann 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.

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

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

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

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

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

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

[10] M. Chowdhury, C.-P. Jeannerod, V. Neiger, E. Schost, and G. Villard. Faster Algorithms for Multivariate Interpolation With Multiplicities and Simultaneous Polynomial Approximations. *IEEE Trans. Inf. Theory*, 61(5):2370–2387, May 2015.

[11] Clément Pernet. *High Performance and Reliable Algebraic Computing*. Nov. 2014. Habilitation.

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

[13] G.-L. Feng and K. K. Tzeng. A Generalization of the Berlekamp-Massey Algorithm for Multisequence Shift-Register Synthesis with Applications to Decoding Cyclic Codes. *IEEE Trans. Inf. Theory*, 37(5):1274–1287, 1991.

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

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

[16] 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.

[17] 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.

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

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

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

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

[22] A. Lenstra. Factoring Multivariate Polynomials over Finite Fields. *J. Comp. Syst. Sc.*, 30(2):235–248, 1985.

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

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

[25] V. Neiger. Fast computation of shifted Popov forms of polynomial matrices via systems of modular polynomial equations. Submitted to ISSAC'16.

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

[27] Z. Olesh and A. Storjohann. The vector rational function reconstruction problem. In *Proc. of WWCA*, pages 137–149, 2006.

[28] G. Schmidt, V. Sidorenko, and M. Bossert. Collaborative Decoding of Interleaved Reed–Solomon Codes and Concatenated Code Designs. *IEEE Trans. Inf. Theory*, 55(7):2991–3012, 2009.

[29] V. Sidorenko and G. Schmidt. A Linear Algebraic Approach to Multisequence Shift-Register Synthesis. *Prob. Inf. Trans.*, 47(2):149–165, 2011.

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

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

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

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

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