Journal of Symbolic Computation 83 (2017) 272–314

Contents lists available at ScienceDirect

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

Computing minimal interpolation bases

Claude-Pierre Jeannerod[a], Vincent Neiger[b], Éric Schost[c], Gilles Villard[d]

[a] Inria, Université de Lyon, Laboratoire LIP (CNRS, Inria, ENS de Lyon, UCBL), Lyon, France
[b] ENS de Lyon, Université de Lyon, Laboratoire LIP (CNRS, Inria, ENS de Lyon, UCBL), Lyon, France
[c] University of Waterloo, David Cheriton School of Computer Science, Waterloo ON, Canada
[d] CNRS, Université de Lyon, Laboratoire LIP (CNRS, Inria, ENS de Lyon, UCBL), Lyon, France

ARTICLE INFO

Article history:
Received 7 December 2015
Accepted 30 April 2016
Available online 15 November 2016

Keywords:
M-Padé approximation
Hermite–Padé approximation
Polynomial interpolation
Order basis
Polynomial matrix

ABSTRACT

We consider the problem of computing univariate polynomial matrices over a field that represent minimal solution bases for a general interpolation problem, some forms of which are the *vector M-Padé approximation problem* in Van Barel and Bultheel (1992) and the *rational interpolation problem* in Beckermann and Labahn (2000). Particular instances of this problem include the bivariate interpolation steps of Guruswami–Sudan hard-decision and Kötter–Vardy soft-decision decodings of Reed–Solomon codes, the multivariate interpolation step of list-decoding of folded Reed–Solomon codes, and Hermite–Padé approximation.

In the mentioned references, the problem is solved using iterative algorithms based on recurrence relations. Here, we discuss a fast, divide-and-conquer version of this recurrence, taking advantage of fast matrix computations over the scalars and over the polynomials. This new algorithm is deterministic, and for computing shifted minimal bases of relations between m vectors of size σ it uses $\mathcal{O}\tilde{}(m^{\omega-1}(\sigma + |\mathbf{s}|))$ field operations, where ω is the exponent of matrix multiplication, and $|\mathbf{s}|$ is the sum of the entries of the input shift $\mathbf{s}$, with $\min(\mathbf{s}) = 0$. This complexity bound improves in particular on earlier algorithms in the case of bivariate interpolation for soft decoding, while matching fastest existing algorithms for simultaneous Hermite–Padé approximation.

E-mail address: vincent.neiger@ens-lyon.fr (V. Neiger).

http://dx.doi.org/10.1016/j.jsc.2016.11.015

1. Introduction

1.1. Context

In this paper, we study fast algorithms for generalizations of classical *Padé approximation* and *polynomial interpolation* problems. Two typical examples of such problems are the following.

Constrained bivariate interpolation. In coding theory, some decoding algorithms rely on solving a bivariate interpolation problem which may be formulated as follows. Given a set of σ points $\{(x_1, y_1), \ldots, (x_\sigma, y_\sigma)\}$ with coordinates in a field $\mathbb{K}$, find a non-zero polynomial $Q \in \mathbb{K}[X, Y]$ of Y-degree less than m satisfying

$$Q(x_1, y_1) = \cdots = Q(x_\sigma, y_\sigma) = 0,$$

as well as a weighted degree constraint. In terms of linear algebra, we interpret this using the $\mathbb{K}$-linear functionals $\ell_1, \ldots, \ell_\sigma$ defined by $\ell_j(Q) = Q(x_j, y_j)$ for polynomials Q in $\mathbb{K}[X, Y]$. Then, given the points, the problem is to find a polynomial Q satisfying the degree constraints and such that $\ell_j(Q) = 0$ for each j. Writing $Q = \sum_{j<m} p_{j+1}(X)Y^j$, in this context, one may actually want to compute a whole basis $\mathbf{P}$ of such interpolants $\mathbf{p} = (p_1, \ldots, p_m)$, and the weighted degree constraint is satisfied through the minimization of some suitably defined degree of $\mathbf{P}$.

Hermite–Padé approximation. Given a vector of m polynomials $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{K}[X]^m$, with coefficients in a field $\mathbb{K}$, and given a target order σ, find another vector of polynomials $\mathbf{p} = (p_1, \ldots, p_m)$ such that

$$p_1 f_1 + \cdots + p_m f_m = 0 \bmod X^\sigma, \qquad (1)$$

with some prescribed degree constraints on $p_1, \ldots, p_m$.

Here as well, one may actually wish to compute a set of such vectors $\mathbf{p}$, forming the rows of a matrix $\mathbf{P}$ over $\mathbb{K}[X]$, which describe a whole basis of solutions. Then, these vectors may not all satisfy the degree constraints, but by requiring that the basis matrix $\mathbf{P}$ minimizes some suitably defined degree, we will ensure that at least one of its rows does (unless the problem has no solution).

Concerning Hermite–Padé approximation, a minimal basis of solutions can be computed in $\mathcal{O}\tilde{~}(m^{\omega-1}\sigma)$ operations in $\mathbb{K}$ (Zhou and Labahn, 2012). Here and hereafter, the soft-O notation $\mathcal{O}\tilde{~}(\cdot)$ indicates that we omit polylogarithmic terms, and the exponent ω is so that we can multiply $m \times m$ matrices over $\mathbb{K}$ in $\mathcal{O}(m^\omega)$ operations in $\mathbb{K}$, the best known bound being $\omega < 2.38$ (Coppersmith and Winograd, 1990; Le Gall, 2014). For constrained bivariate interpolation, assuming that $x_1, \ldots, x_\sigma$ are pairwise distinct, the best known cost bound for computing a minimal basis is $\mathcal{O}\tilde{~}(m^\omega\sigma)$ (Bernstein, 2011; Cohn and Heninger, 2015; Nielsen, 2014); the cost bound $\mathcal{O}\tilde{~}(m^{\omega-1}\sigma)$ was achieved in Chowdhury et al. (2015) with a probabilistic algorithm which outputs only one interpolant satisfying the degree constraints.

Following the work of Van Barel and Bultheel (1992), Beckermann and Labahn (2000), and McEliece's presentation of Kötter's algorithm (McEliece, 2003, Section 7), we adopt a framework that encompasses both examples above, and many other applications detailed in Section 2; we propose a deterministic algorithm for computing a minimal basis of solutions to this general problem.

1.2. Minimal interpolation bases

Consider a field $\mathbb{K}$ and the vector space $\mathfrak{E} = \mathbb{K}^\sigma$, for some positive integer σ; we see its elements as row vectors. Choosing a $\sigma \times \sigma$ matrix $\mathbf{J}$ with entries in $\mathbb{K}$ allows us to make $\mathfrak{E}$ a $\mathbb{K}[X]$-module in the usual manner, by setting $p \cdot \mathbf{e} = \mathbf{e}\, p(\mathbf{J})$, for p in $\mathbb{K}[X]$ and $\mathbf{e}$ in $\mathfrak{E}$. We will call $\mathbf{J}$ the *multiplication matrix* of $(\mathfrak{E}, \cdot)$.

Definition 1.1 *(Interpolant).* Given a vector $\mathbf{E} = (\mathbf{e}_1, \dots, \mathbf{e}_m)$ in $\mathfrak{E}^m$ and a vector $\mathbf{p} = (p_1, \dots, p_m)$ in $\mathbb{K}[X]^m$, we write $\mathbf{p} \cdot \mathbf{E} = p_1 \cdot \mathbf{e}_1 + \cdots + p_m \cdot \mathbf{e}_m \in \mathfrak{E}$. We say that $\mathbf{p}$ is an *interpolant for* $(\mathbf{E}, \mathbf{J})$ if

$$\mathbf{p} \cdot \mathbf{E} = 0. \tag{2}$$

Here, $\mathbf{p}$ is seen as a row vector, and $\mathbf{E}$ is seen as a column vector of m elements of $\mathfrak{E}$: as a matter of notation, $\mathbf{E}$ will often equivalently be seen as an $m \times \sigma$ matrix over $\mathbb{K}$.

Interpolants $\mathbf{p}$ are often called *relations* or *syzygies* of $\mathbf{e}_1, \dots, \mathbf{e}_m$. This notion of interpolants was introduced by Beckermann and Labahn (2000), with the requirement that $\mathbf{J}$ be upper triangular. One of the main results of this paper holds with no assumption on $\mathbf{J}$; for our second main result, we will work under the stronger assumption that $\mathbf{J}$ is a Jordan matrix: it has n Jordan blocks of respective sizes $\sigma_1, \dots, \sigma_n$ and with respective eigenvalues $x_1, \dots, x_n$.

In the latter context, the notion of interpolant directly relates to the one introduced by Van Barel and Bultheel (1992) in terms of $\mathbb{K}[X]$-modules. Indeed, one may identify $\mathfrak{E}$ with the product of residue class rings

$$\mathfrak{F} = \mathbb{K}[X]/(X^{\sigma_1}) \times \cdots \times \mathbb{K}[X]/(X^{\sigma_n}),$$

by mapping a vector $\mathbf{f} = (f_1, \dots, f_n)$ in $\mathfrak{F}$ to the vector $\mathbf{e} \in \mathfrak{E}$ made from the concatenation of the coefficient vectors of $f_1, \dots, f_n$. Then, over $\mathfrak{F}$, the $\mathbb{K}[X]$-module structure on $\mathfrak{E}$ given by $p \cdot \mathbf{e} = \mathbf{e}\, p(\mathbf{J})$ simply becomes

$$p \cdot \mathbf{f} = (p(X + x_1) f_1 \bmod X^{\sigma_1}, \dots, p(X + x_n) f_n \bmod X^{\sigma_n}).$$

Now, if $(\mathbf{e}_1, \dots, \mathbf{e}_m)$ in $\mathfrak{E}^m$ is associated to $(\mathbf{f}_1, \dots, \mathbf{f}_m)$ in $\mathfrak{F}^m$, with $\mathbf{f}_i = (f_{i,1}, \dots, f_{i,n})$ and $f_{i,j}$ in $\mathbb{K}[X]/(X^{\sigma_j})$ for all i, j, the relation $p_1 \cdot \mathbf{e}_1 + \cdots + p_m \cdot \mathbf{e}_m = 0$ means that for all j in $\{1, \dots, n\}$, we have

$$p_1(X + x_j) f_{1,j} + \cdots + p_m(X + x_j) f_{m,j} = 0 \bmod X^{\sigma_j};$$

applying a translation by $-x_j$, this is equivalent to

$$p_1 f_{1,j}(X - x_j) + \cdots + p_m f_{m,j}(X - x_j) = 0 \bmod (X - x_j)^{\sigma_j}. \tag{3}$$

Thus, in terms of vector M-Padé approximation as in Van Barel and Bultheel (1992), $(p_1, \dots, p_m)$ is an interpolant for $(\mathbf{f}_1, \dots, \mathbf{f}_m)$, $x_1, \dots, x_n$, and $\sigma_1, \dots, \sigma_n$.

Both examples above, and many more along the same lines, can be cast into this setting. In the second example above, this is straightforward: we have $n = 1$, $x_1 = 0$, so that the multiplication matrix is the upper shift matrix

$$\mathbf{Z} = \begin{bmatrix} 0 & 1 & & & \\ & \ddots & \ddots & & \\ & & 0 & 1 \\ & & & 0 \end{bmatrix}; \tag{4}$$

it is a nilpotent Jordan block. In the first example, we have $n = \sigma$, and the multiplication matrix is the diagonal matrix

$$\mathbf{D} = \begin{bmatrix} x_1 & & & \\ & x_2 & & \\ & & \ddots & \\ & & & x_\sigma \end{bmatrix}.$$

Then, for p in $\mathbb{K}[X]$ and $\mathbf{e} = [e_1, \ldots, e_\sigma]$ in $\mathfrak{E}$, $p \cdot \mathbf{e}$ is the row vector $[p(x_1)e_1, \ldots, p(x_\sigma)e_\sigma]$. In this case, to solve the interpolation problem, we start from the tuple of bivariate polynomials $(1, Y, \ldots, Y^{m-1})$; their evaluations $\mathbf{E} = (\mathbf{e}_1, \ldots, \mathbf{e}_m)$ in $\mathfrak{E}^m$ are the vectors $\mathbf{e}_i = [y_1^i, \ldots, y_\sigma^i]$ and the relation $\mathbf{p} \cdot \mathbf{E} = 0$ precisely means that $Q(X, Y) = p_1 + p_2 Y + \cdots + p_m Y^{m-1}$ vanishes at all points $\{(x_j, y_j), 1 \leqslant j \leqslant \sigma\}$, where $\mathbf{p} = (p_1, \ldots, p_m)$.

Let us come back to our general context. The set of all interpolants for $(\mathbf{E}, \mathbf{J})$ is a submodule of $\mathbb{K}[X]^m$, which we will denote by $\mathfrak{I}(\mathbf{E}, \mathbf{J})$. Since it contains $\Pi_{\mathbf{J}}(X)\mathbb{K}[X]^m$, where $\Pi_{\mathbf{J}} \in \mathbb{K}[X]$ is the minimal polynomial of $\mathbf{J}$, this submodule is free of rank m (see for example Dummit and Foote, 2004, Chapter 12, Theorem 4).

Definition 1.2 *(Interpolation basis).* Given $\mathbf{E}$ in $\mathfrak{E}^m$ and $\mathbf{J}$ in $\mathbb{K}^{\sigma \times \sigma}$, a matrix $\mathbf{P}$ in $\mathbb{K}[X]^{m \times m}$ is an *interpolation basis for* $(\mathbf{E}, \mathbf{J})$ if its rows form a basis of $\mathfrak{I}(\mathbf{E}, \mathbf{J})$.

In terms of notation, if a matrix $\mathbf{P} \in \mathbb{K}[X]^{k \times m}$ has rows $\mathbf{p}_1, \ldots, \mathbf{p}_k$, we write $\mathbf{P} \cdot \mathbf{E}$ for $(\mathbf{p}_1 \cdot \mathbf{E}, \ldots, \mathbf{p}_k \cdot \mathbf{E}) \in \mathfrak{E}^k$, seen as a column vector. Thus, for $k = m$, if $\mathbf{P}$ is an interpolation basis for $(\mathbf{E}, \mathbf{J})$ then in particular $\mathbf{P} \cdot \mathbf{E} = 0$.

In many situations, one wants to compute an interpolation basis which has sufficiently small degrees: as we will see in Section 2, most previous algorithms compute a basis which is *reduced* with respect to some degree *shift*. In what follows, by shift, we mean a tuple of nonnegative integers which will be used as degree weights on the columns of a polynomial matrix. Before giving a precise definition of shifted minimal interpolation bases, we recall the notions of shifted row degree and shifted reducedness for univariate polynomial matrices; for more details we refer to (Kailath, 1980) and (Zhou, 2012, Chapter 2).

The row degree of a matrix $\mathbf{P} = [p_{i,j}]_{i,j}$ in $\mathbb{K}[X]^{k \times m}$ with no zero row is the tuple $\mathrm{rdeg}(\mathbf{P}) = (d_1, \ldots, d_k) \in \mathbb{N}^k$ with $d_i = \max_j \deg(p_{i,j})$ for all i. For a shift $\mathbf{s} = (s_1, \ldots s_m) \in \mathbb{N}^m$, the diagonal matrix with diagonal entries $X^{s_1}, \ldots, X^{s_m}$ is denoted by $\mathbf{X}^{\mathbf{s}}$, and the $\mathbf{s}$-row degree of $\mathbf{P}$ is $\mathrm{rdeg}_{\mathbf{s}}(\mathbf{P}) = \mathrm{rdeg}(\mathbf{P}\mathbf{X}^{\mathbf{s}})$. Then, the $\mathbf{s}$-leading matrix of $\mathbf{P}$ is the matrix in $\mathbb{K}^{k \times m}$ whose entries are the coefficients of degree zero of $\mathbf{X}^{-\mathrm{rdeg}_{\mathbf{s}}(\mathbf{P})}\mathbf{P}\mathbf{X}^{\mathbf{s}}$, and we say that $\mathbf{P}$ is $\mathbf{s}$-reduced when its $\mathbf{s}$-leading matrix has full rank. In particular, if $\mathbf{P}$ is square (still with no zero row), it is $\mathbf{s}$-reduced if and only if its $\mathbf{s}$-leading matrix is invertible. We note that $\mathbf{P}$ is $\mathbf{s}$-reduced if and only if $\mathbf{P}\mathbf{X}^{\mathbf{s}}$ is $\mathbf{0}$-reduced, where $\mathbf{0} = (0, \ldots, 0)$ is called the *uniform* shift.

Definition 1.3 *(Shifted minimal interpolation basis).* Consider $\mathfrak{E} = \mathbb{K}^\sigma$ and a multiplication matrix $\mathbf{J}$ in $\mathbb{K}^{\sigma \times \sigma}$. Given $\mathbf{E} = (\mathbf{e}_1, \ldots, \mathbf{e}_m)$ in $\mathfrak{E}^m$ and a shift $\mathbf{s} \in \mathbb{N}^m$, a matrix $\mathbf{P} \in \mathbb{K}[X]^{m \times m}$ is said to be an $\mathbf{s}$*-minimal interpolation basis for* $(\mathbf{E}, \mathbf{J})$ if

- $\mathbf{P}$ is an interpolation basis for $(\mathbf{E}, \mathbf{J})$, and
- $\mathbf{P}$ is $\mathbf{s}$-reduced.

We recall that all bases of a free $\mathbb{K}[X]$-module of rank m are unimodularly equivalent: given two bases $\mathbf{A}$ and $\mathbf{B}$, there exists $\mathbf{U} \in \mathbb{K}[X]^{m \times m}$ such that $\mathbf{A} = \mathbf{U}\mathbf{B}$ and $\mathbf{U}$ is unimodular (that is, $\mathbf{U}$ is invertible in $\mathbb{K}[X]^{m \times m}$). Among all the interpolation bases for $(\mathbf{E}, \mathbf{J})$, an $\mathbf{s}$-minimal basis $\mathbf{P}$ has a type of minimal degree property. Indeed, $\mathbf{P}$ is $\mathbf{s}$-reduced if and only if $\mathrm{rdeg}_{\mathbf{s}}(\mathbf{P}) \leqslant \mathrm{rdeg}_{\mathbf{s}}(\mathbf{U}\mathbf{P})$ for any unimodular $\mathbf{U}$; in this inequality, the tuples are first sorted in non-decreasing order and then compared lexicographically. In particular, a row of $\mathbf{P}$ which has minimal $\mathbf{s}$-row degree among the rows of $\mathbf{P}$ also has minimal $\mathbf{s}$-row degree among *all* interpolants for $(\mathbf{E}, \mathbf{J})$.

1.3. Main results

In this article, we propose fast deterministic algorithms that solve Problem 1. Our first main result deals with an arbitrary matrix $\mathbf{J}$, and uses techniques from fast linear algebra.

Taking $\mathbf{J}$ *upper triangular* as in Beckermann and Labahn (2000) would allow us to design a divide-and-conquer algorithm, using the leading and trailing principal submatrices of $\mathbf{J}$ for the recursive calls. However, this assumption alone is not enough to obtain an algorithm with cost quasi-linear in σ, as

Problem 1 *(Minimal interpolation basis).*

Input:

- *the base field* $\mathbb{K}$,
- *the dimensions* m *and* σ,
- *a matrix* $\mathbf{E} \in \mathbb{K}^{m \times \sigma}$,
- *a matrix* $\mathbf{J} \in \mathbb{K}^{\sigma \times \sigma}$,
- *a shift* $\mathbf{s} \in \mathbb{N}^m$.

Output: *an* $\mathbf{s}$*-minimal interpolation basis* $\mathbf{P} \in \mathbb{K}[X]^{m \times m}$ *for* $(\mathbf{E}, \mathbf{J})$.

simply representing $\mathbf{J}$ would require a number of coefficients in $\mathbb{K}$ quadratic in σ. Taking $\mathbf{J}$ a Jordan matrix solves this issue, and is not a strong restriction for applications: it is satisfied in all those we have in mind, which are detailed in Section 2.

If $\mathbf{J} \in \mathbb{K}^{\sigma \times \sigma}$ is a Jordan matrix with n diagonal blocks of respective sizes $\sigma_1, \ldots, \sigma_n$ and with respective eigenvalues $x_1, \ldots, x_n$, we will write it in a compact manner by specifying only those sizes and eigenvalues. Precisely, we will assume that $\mathbf{J}$ is given to us as the form

$$\mathbf{J} = ((x_1, \sigma_{1,1}), \ldots, (x_1, \sigma_{1,r_1}), \ldots, (x_t, \sigma_{t,1}), \ldots, (x_t, \sigma_{t,r_t})), \tag{5}$$

for some pairwise distinct $x_1, \ldots, x_t$, with $r_1 \geqslant \cdots \geqslant r_t$ and $\sigma_{i,1} \geqslant \cdots \geqslant \sigma_{i,r_i}$ for all i; we will say that this representation is *standard*. If $\mathbf{J}$ is given as an arbitrary list $((x_1, \sigma_1), \ldots, (x_n, \sigma_n))$, we can reorder it (and from that, permute the columns of $\mathbf{E}$ accordingly) to bring it to the above form in time $\mathcal{O}(\mathsf{M}(\sigma)\log(\sigma)^3)$ using the algorithm of Bostan et al. (2008, Proposition 12); if $\mathbb{K}$ is equipped with an order, and if we assume that comparisons take unit time, it is of course enough to sort the x_i's. Here, $\mathsf{M}(\cdot)$ is a multiplication time function for $\mathbb{K}[X]$: polynomials of degree at most d in $\mathbb{K}[X]$ can be multiplied using $\mathsf{M}(d)$ operations in $\mathbb{K}$, and $\mathsf{M}(\cdot)$ satisfies the super-linearity properties of von zur Gathen and Gerhard (2013, Chapter 8). It follows from the algorithm of Cantor and Kaltofen (1991) that $\mathsf{M}(d)$ can be taken in $\mathcal{O}(d\log(d)\log(\log(d)))$.

Adding a constant to every entry of $\mathbf{s}$ does not change the notion of $\mathbf{s}$-reducedness, and thus does not change the output matrix $\mathbf{P}$ of Problem 1; in particular, one may ensure that $\min(\mathbf{s}) = 0$ without loss of generality. The shift $\mathbf{s}$, as a set of degree weights on the columns of $\mathbf{P}$, naturally affects how the degrees of the entries of $\mathbf{P}$ are distributed. Although no precise degree profile of $\mathbf{P}$ can be stated in general, we do have a global control over the degrees in $\mathbf{P}$, as showed in the following results; in these statements, for a shift $\mathbf{s}$, we write $|\mathbf{s}|$ to denote the quantity $|\mathbf{s}| = s_1 + \cdots + s_m$.

We start with the most general result in this paper, where we make no assumption on $\mathbf{J}$. In this case, we obtain an algorithm whose cost is essentially that of fast linear algebra over $\mathbb{K}$. The output of this algorithm has an extra uniqueness property: it is in *Popov form*; we refer the reader to Section 7 or (Beckermann et al., 2006) for a definition.

Theorem 1.4. *There is a deterministic algorithm which solves Problem 1 using*

$$\begin{aligned} &\mathcal{O}(\sigma^{\omega}(\lceil m/\sigma \rceil + \log(\sigma))) && \text{if } \omega > 2 \\ &\mathcal{O}(\sigma^{2}(\lceil m/\sigma \rceil + \log(\sigma))\log(\sigma)) && \text{if } \omega = 2 \end{aligned}$$

operations in $\mathbb{K}$ *and returns the unique* $\mathbf{s}$*-minimal interpolation basis for* $(\mathbf{E}, \mathbf{J})$ *which is in* $\mathbf{s}$*-Popov form. Besides, the sum of the column degrees of this basis is at most* σ.

In the usual case where $m = \mathcal{O}(\sigma)$, the cost is thus $\mathcal{O}\tilde{~}(\sigma^{\omega})$; this is to be compared with the algorithm of Beckermann and Labahn (2000), which we discuss in Section 2.

Our second main result deals with the case of $\mathbf{J}$ in Jordan canonical form, for which we obtain a cost bound that is quasi-linear with respect to σ.

Theorem 1.5. *Assuming that* $\mathbf{J} \in \mathbb{K}^{\sigma \times \sigma}$ *is a Jordan matrix, given by a standard representation, there is a deterministic algorithm which solves Problem 1 using*

$$\mathcal{O}(m^{\omega-1}\mathsf{M}(\sigma)\log(\sigma)\log(\sigma/m) + m^{\omega-1}\mathsf{M}(\xi)\log(\xi/m)) \qquad \text{if } \omega > 2$$

$$\mathcal{O}(m\mathsf{M}(\sigma)\log(\sigma)\log(\sigma/m)\log(m)^3 + m\mathsf{M}(\xi)\log(\xi/m)\log(m)^2) \qquad \text{if } \omega = 2$$

operations in $\mathbb{K}$, *where* $\xi = |\mathbf{s} - \min(\mathbf{s})|$. *Besides, the sum of the row degrees of the computed* $\mathbf{s}$*-minimal interpolation basis is at most* $\sigma + \xi$.

The reader interested in the logarithmic factors should refer to the more precise cost bound in Proposition 3.1. Masking logarithmic factors, this cost bound is $\mathcal{O}\tilde{}(m^{\omega-1}(\sigma + \xi))$. We remark that the bound on the output row degree implies that the size of the output matrix $\mathbf{P}$ is $\mathcal{O}(m(\sigma + \xi))$, where by size we mean the number of coefficients of $\mathbb{K}$ needed to represent this matrix.

We are not aware of a previous cost bound for the general question stated in Problem 1 that would be similar to our result; we give a detailed comparison with several previous algorithms and discuss useful particular cases in Section 2.

1.4. Overview of the algorithms

To deal with an arbitrary matrix $\mathbf{J}$, we rely on a linear algebra approach presented in Section 7, using a linearization framework that is classical for this kind of problems (Kailath, 1980). Our algorithm computes the rank profile of a block Krylov matrix using techniques that are reminiscent of the algorithm of Keller-Gehrig (1985); this framework also allows us to derive a bound on the sum of the row degrees of shifted minimal interpolation bases. Section 7 is the last section of this paper; it is the only section where we make no assumption on $\mathbf{J}$, and it does not use results from other parts of the paper.

We give in Section 3 a divide-and-conquer algorithm for the case of a matrix $\mathbf{J}$ in Jordan canonical form. The idea is to use a Knuth–Schönhage-like half-gcd approach (Knuth, 1970; Schönhage, 1971; Brent et al., 1980), previously carried over to the specific case of simultaneous Hermite–Padé approximation in Beckermann and Labahn (1994), Giorgi et al. (2003). This approach consists in reducing a problem in size σ to a first sub-problem in size $\sigma/2$, the computation of the so-called *residual*, a second sub-problem in size $\sigma/2$, and finally a recombination of the results of both sub-problems via polynomial matrix multiplication. The shift to be used in the second recursive call is essentially the $\mathbf{s}$-row degree of the outcome of the first recursive call.

The main difficulty is to control the sizes of the interpolation bases that are obtained recursively. The bound we rely on, as stated in our main theorems, depends on the input shift. In our algorithm, we cannot make any assumption on the shifts that will appear in recursive calls, since they depend on the degrees of the previously computed bases. Hence, even in the case of a uniform input shift for which the output basis is of size $\mathcal{O}(m\sigma)$, there may be recursive calls with an unbalanced shift, which may output bases that have large size.

Our workaround is to perform all recursive calls with the uniform shift $\mathbf{s} = \mathbf{0}$, and resort to a change of shift that will be studied in Section 5; this strategy is an alternative to the linearization approach that was used by Zhou and Labahn (2012) in the specific case of simultaneous Hermite–Padé approximation. We note that our change of shift uses an algorithm in Zhou et al. (2012), which itself relies on simultaneous Hermite–Padé approximation; with the dimensions needed here, this approximation problem is solved efficiently using the algorithm of Giorgi et al. (2003), without resorting to (Zhou and Labahn, 2012).

Another difficulty is to deal with instances where σ is small. Our bound on the size of the output states that when $\sigma \leqslant m$ and the shift is uniform, the average degree of the entries of a minimal interpolation basis is at most 1. Thus, in this case our focus is not anymore on using fast polynomial arithmetic but rather on exploiting efficient linear algebra over $\mathbb{K}$: the divide-and-conquer process stops when reaching $\sigma \leqslant m$, and invokes instead the algorithm based on linear algebra proposed in Section 7.

The last ingredient is the fast computation of the residual, that is, a matrix in $\mathbb{K}^{m\times\sigma/2}$ for restarting the process after having found a basis $\mathbf{P}^{(1)}$ for the first sub-problem of size $\sigma/2$. This boils down to computing $\mathbf{P}^{(1)}\cdot\mathbf{E}$ and discarding the first $\sigma/2$ columns, which are known to be zero. In Section 6, we design a general procedure for computing this kind of product, using Hermite interpolation and evaluation to reduce it to multiplying polynomial matrices.

Concerning the multiplication of the bases obtained recursively, to handle the fact that they may have unbalanced row degrees, we use the approach in Zhou et al. (2012, Section 3.6); we give a detailed algorithm and cost analysis in Section 4.

2. Applications and comparisons with previous work

In this section, we review and expand the scope of the examples described in the introduction, and we compare our results for these examples to previous work. *For ease of comparison, in all this section we consider the case $\omega > 2$.*

2.1. General case

In this paragraph, we consider the general Problem 1, without assuming that $\mathbf{J}$ is a Jordan matrix. The only previous work that we are aware of is Beckermann and Labahn (2000), where it is still assumed that $\mathbf{J}$ is upper triangular. This assumption allows one to use an iteration on the columns of $\mathbf{E}$, combining Gaussian elimination with multiplication by monic polynomials of degree 1 to build the basis $\mathbf{P}$: after i iterations, $\mathbf{P}$ is an $\mathbf{s}$-minimal interpolation basis for the first i columns of $\mathbf{E}$ and the $i\times i$ leading principal submatrix of $\mathbf{J}$. This algorithm uses $\mathcal{O}(m\sigma^4)$ operations and returns a basis in $\mathbf{s}$-Popov form. We note that in this context the mere representation of $\mathbf{J}$ uses $\Theta(\sigma^2)$ elements.

As a comparison, the algorithm of Theorem 1.4 computes an interpolation basis for $(\mathbf{E},\mathbf{J})$ in $\mathbf{s}$-Popov form for *any* matrix $\mathbf{J}$ in $\mathbb{K}^{\sigma\times\sigma}$. For any shift $\mathbf{s}$, this algorithm uses $\mathcal{O}(\sigma^\omega\log(\sigma))$ operations when $m\in\mathcal{O}(\sigma)$ and $\mathcal{O}(\sigma^{\omega-1}m+\sigma^\omega\log(\sigma))$ operations when $\sigma\in\mathcal{O}(m)$.

2.2. M-Padé approximation

We continue with the case of a matrix $\mathbf{J}$ in Jordan canonical form. As pointed out in the introduction, Problem 1 can be formulated in this case in terms of polynomial equations and corresponds to an M-Padé approximation problem; this problem was studied in Lübbe (1983), Beckermann (1990) and named after the work of Mahler, including in particular (Mahler, 1968). Indeed, up to applying a translation in the input polynomials, the problem stated in (3) can be rephrased as follows.

Problem 2 *(M-Padé approximation).*

Input:

- *points* $x_1,\ldots,x_n$ *in* $\mathbb{K}$,
- *integers* $\sigma_1\geqslant\cdots\geqslant\sigma_n>0$,
- *matrix* $\mathbf{F}$ *in* $\mathbb{K}[X]^{m\times n}$ *with its* j*-th column* $\mathbf{F}_{*,j}$ *of degree* $<\sigma_j$,
- *shift* $\mathbf{s}\in\mathbb{N}^m$.

Output: *a matrix* $\mathbf{P}$ *in* $\mathbb{K}[X]^{m\times m}$ *such that*

- *the rows of* $\mathbf{P}$ *form a basis of the* $\mathbb{K}[X]$*-module*

$$\{\mathbf{p}\in\mathbb{K}[X]^{1\times m}\ \mid\ \mathbf{pF}_{*,j}=0 \bmod (X-x_j)^{\sigma_j} \text{ for each } j\},$$

- $\mathbf{P}$ *is* $\mathbf{s}$*-reduced.*

Here, $\sigma=\sigma_1+\cdots+\sigma_n$. Previous work on this particular problem includes (Beckermann, 1992; Van Barel and Bultheel, 1992); note that in Beckermann (1992), the input consists of a single column $\mathbf{F}$ in $\mathbb{K}[X]^{m\times 1}$ of degree less than σ: to form the input of our problem, we compute $\hat{\mathbf{F}}=$

$[\mathbf{F} \bmod (X-x_1)^{\sigma_1} | \cdots | \mathbf{F} \bmod (X-x_n)^{\sigma_n}]$. The algorithms in these references have a cost of $\mathcal{O}(m^2\sigma^2)$ operations, which can be lowered to $\mathcal{O}(m\sigma^2)$ if one computes only the small degree rows of an $\mathbf{s}$-minimal basis (gradually discarding the rows of degree more than, say, $2\sigma/m$ during the computation). To the best of our knowledge, no algorithm with a cost quasi-linear in σ has been given in the literature.

2.3. Hermite–Padé approximation

Specializing the discussion of the previous paragraph to the case where all x_i are zero, we obtain the following important particular case.

Problem 3 *(Simultaneous Hermite–Padé approximation).*

Input:

- *integers* $\sigma_1 \geqslant \cdots \geqslant \sigma_n > 0$,
- *matrix* $\mathbf{F}$ *in* $\mathbb{K}[X]^{m\times n}$ *with its* j*-th column* $\mathbf{F}_{*,j}$ *of degree* $< \sigma_j$,
- *shift* $\mathbf{s} \in \mathbb{N}^m$.

Output: *a matrix* $\mathbf{P}$ *in* $\mathbb{K}[X]^{m\times m}$ *such that*

- *the rows of* $\mathbf{P}$ *form a basis of the* $\mathbb{K}[X]$*-module*

$$\{\mathbf{p} \in \mathbb{K}[X]^{1\times m} \mid \mathbf{p}\mathbf{F}_{*,j} = 0 \bmod X^{\sigma_j} \text{ for each } j\},$$

- $\mathbf{P}$ *is* $\mathbf{s}$*-reduced.*

Here, $\sigma = \sigma_1 + \cdots + \sigma_m$. Our main result says that there is an algorithm which solves Problem 3 using $\mathcal{O}(m^{\omega-1}\mathsf{M}(\sigma)\log(\sigma)\log(\sigma/m) + m^{\omega-1}\mathsf{M}(\xi)\log(\xi/m))$ operations in $\mathbb{K}$, where $\xi = |\mathbf{s} - \min(\mathbf{s})|$. As we will see, slightly faster algorithms for this problem exist in the literature, but they do not cover the same range of cases as we do; to the best of our knowledge, for instance, most previous work on this problem dealt with the case $\sigma_1 = \cdots = \sigma_n = \sigma/n$ and $n \leqslant m$.

First algorithms with a cost quadratic in σ were given in Sergeyev (1987), Paszkowski (1987) in the case of Hermite–Padé approximation ($n = 1$), assuming a type of genericity of $\mathbf{F}$ and outputting a single approximant $\mathbf{p} \in \mathbb{K}[X]^{1\times m}$ which satisfies some prescribed degree constraints. For $n = 1$, Van Barel and Bultheel (1991) propose an algorithm which uses $\mathcal{O}(m^2\sigma^2)$ operations to compute a $\mathbf{s}$-minimal basis of approximants for $\mathbf{F}$ at order σ, for any $\mathbf{F}$ and $\mathbf{s}$. This result was extended by Beckermann and Labahn (1994) to the case of any $n \leqslant m$, with the additional remark that the cost bound is $\mathcal{O}(m\sigma^2)$ if one restricts to computing only the rows of small degree of an $\mathbf{s}$-minimal basis.

In Beckermann and Labahn (1994), the authors also propose a divide-and-conquer algorithm using $\mathcal{O}\tilde{~}(m^\omega\sigma)$ operations in $\mathbb{K}$; the base case of the recursion deals with the constant coefficient of a single column of the input $\mathbf{F}$. Then, Giorgi et al. (2003) follow a similar divide-and-conquer approach, introducing a base case which has matrix dimensions $m \times n$ and is solved efficiently by means of linear algebra over $\mathbb{K}$; this yields an algorithm with the cost bound $\mathcal{O}(m^\omega\mathsf{M}(\sigma/n)\log(\sigma/n))$, which is particularly efficient when $n \in \Theta(m)$. On the other hand, in the case of Hermite–Padé approximation ($n = 1$) it was noticed by Lecerf (2001) that the cost bound $\mathcal{O}\tilde{~}(m^\omega\sigma)$ is pessimistic, at least when some type of genericity is assumed concerning the input $\mathbf{F}$, and that in this case there is hope to achieve $\mathcal{O}\tilde{~}(m^{\omega-1}\sigma)$.

This cost bound was then obtained by Storjohann (2006), for computing only the small degree rows of an $\mathbf{s}$-minimal basis, via a reduction of the case of small n and order σ to a case with larger column dimension $n' \approx m$ and smaller order $\sigma' \approx n\sigma/m$. This was exploited to compute a full $\mathbf{s}$-minimal basis using $\mathcal{O}(m^\omega\mathsf{M}(\sigma/m)\log(\sigma/n))$ operations (Zhou and Labahn, 2012), under the assumption that either $|\mathbf{s} - \min(\mathbf{s})| \in \mathcal{O}(\sigma)$ or $|\max(\mathbf{s}) - \mathbf{s}| \in \mathcal{O}(\sigma)$ (which both imply that an $\mathbf{s}$-minimal basis has size $\mathcal{O}(m\sigma)$). We note that this algorithm is faster than ours by a logarithmic factor, and that our result does not cover the second assumption on the shift with a similar cost bound; on the other hand, we handle situations that are not covered by that result, when for instance all σ_i's are not equal.

Zhou (2012, Section 3.6) gives a fast algorithm for the case $n \leqslant \sigma \leqslant m$, for the uniform shift and $\sigma_1 = \cdots = \sigma_n = \sigma/n$. For an input of dimensions $m \times n$, the announced cost bound is $\mathcal{O}\tilde{~}(m^{\omega-1}\sigma)$; a more precise analysis shows that the cost is $\mathcal{O}(\sigma^{\omega-1}m + \sigma^{\omega}\log(\sigma/n))$. For this particular case, there is no point in using our divide-and-conquer algorithm, since the recursion stops at $\sigma \leqslant m$; our general algorithm in Section 7 handles these situations, and its cost (as given in Proposition 7.1 with $\delta = \sigma/n$) is the same as that of Zhou's algorithm in this particular case. In addition, this cost bound is valid for any shift $\mathbf{s}$ and the algorithm returns a basis in $\mathbf{s}$-Popov form.

2.4. Multivariate interpolation

In this subsection, we discuss how our algorithm can be used to solve bivariate interpolation problems, such as those appearing in the list-decoding (Sudan, 1997; Guruswami and Sudan, 1998) and the soft-decoding (Kötter and Vardy, 2003a, Sec. III) of Reed–Solomon codes; as well as multivariate interpolation problems, such as those appearing in the list-decoding of folded Reed–Solomon codes (Guruswami and Rudra, 2008) and in robust Private Information Retrieval (Devet et al., 2012). Our contribution leads to the best known cost bound we are aware of for the interpolation steps of the list-decoding and soft-decoding of Reed–Solomon codes: this is detailed in Subsections 2.5 and 2.6.

In those problems, we have r new variables $Y = Y_1, \ldots, Y_r$ and we want to find a multivariate polynomial $Q(X, Y)$ which vanishes at some given points $\{(x_k, y_k)\}_{1 \leqslant k \leqslant p}$ with prescribed *supports* $\{\mu_k\}_{1 \leqslant k \leqslant p}$. In this context, given a point $(x, y) \in \mathbb{K}^{r+1}$ and an exponent set $\mu \subset \mathbb{N}^{r+1}$, we say that Q *vanishes at* (x, y) *with support* μ if the shifted polynomial $Q(X + x, Y + y)$ has no monomial with exponent in μ. Besides, the solution $Q(X, Y)$ should also have sufficiently small *weighted degree* $\deg_X(Q(X, X^{\mathbf{w}}Y))$ for some given weight $\mathbf{w} \in \mathbb{N}^r$.

To make this fit in our framework, we first require that we are given an exponent set $\Gamma \subseteq \mathbb{N}^r$ such that any monomial $X^i Y^j$ appearing in a solution $Q(X, Y)$ should satisfy $j \in \Gamma$. Then, we can identify $Q(X, Y) = \sum_{j \in \Gamma} p_j(X) Y^j$ with $\mathbf{p} = [p_j]_{j \in \Gamma} \in \mathbb{K}[X]^{1 \times m}$, where m is the cardinality of Γ. We will show below how to construct matrices $\mathbf{E}$ and $\mathbf{J}$ such that solutions $Q(X, Y)$ to those multivariate interpolation problems correspond to interpolants $\mathbf{p} = [p_j]_{j \in \Gamma}$ for $(\mathbf{E}, \mathbf{J})$ that have sufficiently small shifted degree.

We also require that each considered support μ satisfies

$$\text{if } (i, j) \in \mu \text{ and } i > 0, \text{ then } (i - 1, j) \in \mu. \tag{6}$$

Then, the set

$$\mathfrak{I}_{\mathrm{int}} = \left\{ \mathbf{p} = (p_j)_{j \in \Gamma} \in \mathbb{K}[X]^{1 \times m} \;\middle|\; \sum_{j \in \Gamma} p_j(X) Y^j \text{ vanishes at } (x_k, y_k) \text{ with support } \mu_k \text{ for } 1 \leqslant k \leqslant p \right\}$$

is a $\mathbb{K}[X]$-module, as can be seen from the equality

$$(XQ)(X + x, Y + y) = (X + x) Q(X + x, Y + y). \tag{7}$$

Finally, as before, we assume that we are given a shift $\mathbf{s}$ such that we are looking for polynomials of small $\mathbf{s}$-row degree in $\mathfrak{I}_{\mathrm{int}}$. In this context, the shift is often derived from a weighted degree condition which states that, given some input weights $\mathbf{w} = (w_1, \ldots, w_r) \in \mathbb{N}^r$, the degree $\deg_X(Q(X, X^{w_1}Y_1, \cdots, X^{w_r}Y_r))$ should be sufficiently small. Since $Q(X, X^{w_1}Y_1, \cdots, X^{w_r}Y_r) = \sum_{j \in \Gamma} X^{w_1 j_1 + \cdots + w_r j_r} p_j(X) Y^j$, this $\mathbf{w}$-weighted degree of Q is exactly the $\mathbf{s}$-row degree of $\mathbf{p}$ for the shift $\mathbf{s} = (w_1 j_1 + \cdots + w_r j_r)_{(j_1, \ldots, j_r) \in \Gamma}$. An $\mathbf{s}$-reduced basis of $\mathfrak{I}_{\mathrm{int}}$ contains a row of minimal $\mathbf{s}$-row degree among all $\mathbf{p} \in \mathfrak{I}_{\mathrm{int}}$; we note that in some applications, for example in robust Private Information Retrieval (Devet et al., 2012), it is important to return a whole basis of solutions and not only a small degree one.

Problem 4 *(Constrained multivariate interpolation).*

Input:

- *number of variables* $r > 0$,
- *exponent set* $\Gamma \subset \mathbb{N}^r$ *of cardinality* m,
- *pairwise distinct points* $\{(x_k, y_k) \in \mathbb{K}^{r+1}, 1 \leqslant k \leqslant p\}$ *with* $x_1, \ldots, x_p \in \mathbb{K}$ *and* $y_1, \ldots, y_p$ *in* $\mathbb{K}^r$,
- *supports* $\{\mu_k \subset \mathbb{N}^{r+1}, 1 \leqslant k \leqslant p\}$ *where each* μ_k *satisfies* (6),
- *shift* $\mathbf{s} \in \mathbb{N}^m$.

Output: *a matrix* $\mathbf{P} \in \mathbb{K}[X]^{m\times m}$ *such that*

- *the rows of* $\mathbf{P}$ *form a basis of the* $\mathbb{K}[X]$*-module* $\mathfrak{I}_{\mathrm{int}}$,
- $\mathbf{P}$ *is* $\mathbf{s}$*-reduced.*

This problem can be embedded in our framework as follows. We consider $\mathfrak{M} = \mathbb{K}[X, Y_1, \ldots, Y_r]$ and the $\mathbb{K}$-linear functionals $\{\ell_{i,j,k} : \mathfrak{M} \to \mathbb{K}, (i, j) \in \mu_k, 1 \leqslant k \leqslant p\}$, where $\ell_{i,j,k}(Q)$ is the coefficient of $X^i Y^j$ in $Q(X + x_k, Y + y_k)$. These functionals are linearly independent, and the intersection $\mathfrak{K}$ of their kernels is the $\mathbb{K}[X]$-module of polynomials in $\mathfrak{M}$ vanishing at (x_k, y_k) with support μ_k for all k. The quotient $\mathfrak{M}/\mathfrak{K}$ is a $\mathbb{K}$-vector space of dimension $\sigma = \sigma_1 + \cdots + \sigma_p$ where $\sigma_k = \#\mu_k$; it is thus isomorphic to $\mathfrak{E} = \mathbb{K}^\sigma$, with a basis of the dual space given by the functionals $\ell_{i,j,k}$.

Our assumption on the supports μ_k implies that $\mathfrak{M}/\mathfrak{K}$ is a $\mathbb{K}[X]$-module; we now describe the corresponding multiplication matrix $\mathbf{J}$. For a given k, let us order the functionals $\ell_{i,j,k}$ in such a way that, for any (i, j) such that $(i + 1, j)$ is in μ_k, the successor of $\ell_{i,j,k}$ is $\ell_{i+1,j,k}$. Equation (7) implies that

$$\ell_{i,j,k}(XQ) = \ell_{i-1,j,k}(Q) + x_k \ell_{i,j,k}(Q) \tag{8}$$

holds for all Q, all $k \in \{1, \ldots, p\}$, and all $(i, j) \in \mu_k$ with $i > 0$.

Hence, $\mathbf{J}$ is block diagonal with diagonal blocks $\mathbf{J}_1, \ldots, \mathbf{J}_p$, where $\mathbf{J}_k$ is a $\sigma_k \times \sigma_k$ Jordan matrix with only eigenvalue x_k and block sizes given by the support μ_k. More precisely, denoting $\Lambda_k = \{j \in \mathbb{N}^r \mid (i, j) \in \mu_k \text{ for some } i\}$ and $\sigma_{k,j} = \max\{i \in \mathbb{N} \mid (i, j) \in \mu_k\}$ for each $j \in \Lambda_k$, we have the disjoint union

$$\mu_k = \bigcup_{j \in \Lambda_k} \left\{(i, j), 0 \leqslant i \leqslant \sigma_{k,j}\right\}.$$

Then, $\mathbf{J}_k$ is block diagonal with $\#\Lambda_k$ blocks: to each $j \in \Lambda_k$ corresponds a $\sigma_{k,j} \times \sigma_{k,j}$ Jordan block with eigenvalue x_k. It is reasonable to consider $x_1, \ldots, x_p$ ordered as we would like for a standard representation of $\mathbf{J}$. For example, in problems coming from coding theory, these points are part of the code itself, so the reordering can be done as a pre-computation as soon as the code is fixed.

To complete the reduction to Problem 1, it remains to construct $\mathbf{E}$. For each exponent $\gamma \in \Gamma$ we consider the monomial Y^γ and take its image in $\mathfrak{M}/\mathfrak{K}$: this is the vector $\mathbf{e}_\gamma \in \mathfrak{E}$ having for entries the evaluations of the functionals $\ell_{i,j,k}$ at Y^γ. Let then $\mathbf{E}$ be the matrix in $\mathbb{K}^{m\times\sigma}$ with rows $(\mathbf{e}_\gamma)_{\gamma\in\Gamma}$: our construction shows that a row $\mathbf{p} = (p_\gamma)_{\gamma\in\Gamma} \in \mathbb{K}[X]^{1\times m}$ is in $\mathfrak{I}_{\mathrm{int}}$ if and only if it is an interpolant for $(\mathbf{E}, \mathbf{J})$.

To make this reduction to Problem 1 efficient, we make the assumption that the exponent sets Γ and μ_k are *stable under division*: this means that if $j \in \Gamma$ then all j' such that $j' \leqslant j$ (for the product order on $\mathbb{N}^r$) belong to Γ; and if (i, j) is in μ, then all (i', j') such that $(i', j') \leqslant (i, j)$ (for the product order on $\mathbb{N}^{r+1}$) belong to μ_k. This assumption is satisfied in the applications detailed below; besides, using the straightforward extension of (8) to multiplication by $Y_1, \ldots, Y_r$, it allows us to compute all entries $\ell_{i,j,k}(Y^\gamma)$ of the matrix $\mathbf{E}$ inductively in $\mathcal{O}(m\sigma)$, which is negligible compared to the cost of solving the resulting instance of Problem 1. (As a side note, we remark that this assumption also implies that $\mathfrak{K}$ is a zero-dimensional ideal of $\mathfrak{M}$.)

Proposition 2.1. *Assuming that* Γ *and* $\mu_1, \ldots, \mu_p$ *are stable under division, there is an algorithm which solves Problem 4 using*

$$\mathcal{O}(m^{\omega-1}\mathsf{M}(\sigma)\log(\sigma)\log(\sigma/m)+m^{\omega-1}\mathsf{M}(\xi)\log(\xi/m)) \quad \textit{if } \omega>2$$
$$\mathcal{O}(m\mathsf{M}(\sigma)\log(\sigma)\log(\sigma/m)\log(m)^3+m\mathsf{M}(\xi)\log(\xi/m)\log(m)^2) \quad \textit{if } \omega=2$$

operations in $\mathbb{K}$, *where* $\sigma=\#\mu_1+\cdots+\#\mu_p$ *and* $\xi=|\mathbf{s}-\min(\mathbf{s})|$.

2.5. List-decoding of Reed–Solomon codes

In the algorithms of Sudan (1997), Guruswami and Sudan (1999), the bivariate interpolation step deals with Problem 4 with $r=1$, $\Gamma=\{0,\ldots,m-1\}$, pairwise distinct points $x_1,\ldots,x_p$, and $\mu_k=\{(i,j)\mid i+j<b\}$ for all k; b is the *multiplicity parameter* and $m-1$ is the *list-size parameter*. As explained above, the shift $\mathbf{s}$ takes the form $\mathbf{s}=(0,w,2w,\ldots,(m-1)w)$; here $w+1$ is the message length of the considered Reed–Solomon code and p is its block length.

We will see below, in the more general interpolation step of the soft-decoding, that $|\mathbf{s}|\in\mathcal{O}(\sigma)$. As a consequence, Proposition 2.1 states that this bivariate interpolation step can be performed using $\mathcal{O}(m^{\omega-1}\mathsf{M}(\sigma)\log(\sigma)\log(\sigma/m))$ operations. To our knowledge, the best previously known cost bound for this problem is $\mathcal{O}(m^{\omega-1}\mathsf{M}(\sigma)\log(\sigma)^2)$ and was obtained using a randomized algorithm (Chowdhury et al., 2015, Corollary 14). In contrast, Algorithm 1 in this paper makes no random choices. The algorithm in Chowdhury et al. (2015) uses fast structured linear algebra, following an approach studied in Olshevsky and Shokrollahi (1999), Roth and Ruckenstein (2000), Zeh et al. (2011). Restricting to deterministic algorithms, the best previously known cost bound is $\mathcal{O}(m^{\omega}\mathsf{M}(\sigma/b)(\log(m)^2+\log(\sigma/b)))$ (Bernstein, 2011; Cohn and Heninger, 2015), where $b<m$ is the multiplicity parameter mentioned above. This is obtained by first building a known $m\times m$ interpolation basis with entries of degree at most σ/b, and then using fast deterministic reduction of polynomial matrices (Gupta et al., 2012); other references on this approach include (Alekhnovich, 2002; Lee and O'Sullivan, 2008; Beelen and Brander, 2010).

A previous divide-and-conquer algorithm can be found in Nielsen (2014). The recursion is on the number of points p, and using fast multiplication of the bases obtained recursively, this algorithm has cost bound $\mathcal{O}(m^2\sigma b)+\mathcal{O}\tilde{}(m^{\omega}\sigma/b)$ (Nielsen, 2014, Proposition 3). In this reference, the bases computed recursively are allowed to have size as large as $\Theta(m^2\sigma/b)$, with $b<m$.

For a more detailed perspective of the previous work on this specific problem, the reader may for instance refer to the cost comparisons in the introductive sections of Beelen and Brander (2010) and Chowdhury et al. (2015).

2.6. Soft-decoding of Reed–Solomon codes

In Kötter and Vardy's algorithm, the so-called soft-interpolation step (Kötter and Vardy, 2003a, Section III) is Problem 4 with $r=1$, $\Gamma=\{0,\ldots,m-1\}$, and $\mathbf{s}=(0,w,\ldots,(m-1)w)$. The points $x_1,\ldots,x_p$ are not necessarily pairwise distinct, and to each x_k for $k\in\{1,\ldots,p\}$ is associated a multiplicity parameter b_k and a corresponding support $\mu_k=\{(i,j)\mid i+j<b_k\}$. Then, $\sigma=\sum_{1\leqslant k\leqslant p}\binom{b_k+1}{2}$ is called the *cost* (Kötter and Vardy, 2003a, Section III), since it corresponds to the number of linear equations that one obtains in the straightforward linearization of the problem.

In this context, one chooses for m the smallest integer such that the number of linear unknowns in the linearized problem is more than σ. This number of unknowns being directly linked to $|\mathbf{s}|$, this leads to $|\mathbf{s}|\in\mathcal{O}(\sigma)$, which can be proven for example using (Kötter and Vardy, 2003a, Lemma 1 and Equations (10) and (11)). Thus, our algorithm solves the soft-interpolation step using $\mathcal{O}(m^{\omega-1}\mathsf{M}(\sigma)\log(\sigma)\log(\sigma/m))$ operations in $\mathbb{K}$, which is to our knowledge the best known cost bound to solve this problem.

Iterative algorithms, now often referred to as *Kötter's algorithm* in coding theory, were given in Kötter (1996), Nielsen and Høholdt (2000); one may also refer to the general presentation of this solution in McEliece (2003, Section 7). These algorithms use $\mathcal{O}(m\sigma^2)$ operations in $\mathbb{K}$ (for the considered input shifts which satisfy $|\mathbf{s}|\in\mathcal{O}(\sigma)$). We showed above how the soft-interpolation step can be reduced to a specific instance of M-Padé approximation (Problem 2): likewise, one may remark that the algorithms in Nielsen and Høholdt (2000), McEliece (2003) are closely linked to those in Beckermann (1992), Van Barel and Bultheel (1992), Beckermann and Labahn (2000). In particular, up to the interpretation of row vectors in $\mathbb{K}[X]^{m\times m}$ as bivariate polynomials of degree less than m in Y, they use the same recurrence. Our recursive Algorithm 2 is a fast divide-and-conquer version of these algorithms.

An approach based on polynomial matrix reduction was developed in Alekhnovich (2002), Alekhnovich (2005) and in Lee and O'Sullivan (2006). It consists first in building an interpolation basis, that is, a basis $\mathbf{B} \in \mathbb{K}[X]^{m\times m}$ of the $\mathbb{K}[X]$-module $\mathfrak{I}_{\mathrm{int}}$, and then in reducing the basis for the given shift $\mathbf{s}$ to obtain an $\mathbf{s}$-minimal interpolation basis. The maximal degree in $\mathbf{B}$ is

$$\delta = \sum_{1\leqslant k\leqslant p} \frac{\max\{b_i \mid 1\leqslant i\leqslant p \text{ and } x_i = x_k\}}{\#\{1\leqslant i\leqslant p \mid x_i = x_k\}};$$

one can check using $b_k < m$ for all k that

$$\sigma = \sum_{1\leqslant k\leqslant p} \frac{b_k(b_k+1)}{2} \leqslant \frac{m}{2} \sum_{1\leqslant k\leqslant p} b_k \leqslant \frac{m\delta}{2}.$$

Using the fast deterministic reduction algorithm in Gupta et al. (2012), this approach has cost bound $\mathcal{O}(m^\omega \mathsf{M}(\delta)(\log(m)^2 + \log(\delta)))$; the cost bound of our algorithm is thus smaller by a factor $\mathcal{O}^\sim(m\delta/\sigma)$.

In Zeh (2013, Section 5.1) the so-called key equations commonly used in the decoding of Reed–Solomon codes were generalized to this soft-interpolation step. It was then showed in Chowdhury et al. (2015) how one can efficiently compute a solution to these equations using fast structured linear algebra. In this approach, the set of points $\{(x_k, y_k), 1 \leqslant k \leqslant p\}$ is partitioned as $\mathcal{P}_1 \cup \cdots \cup \mathcal{P}_q$, where in each $\mathcal{P}_h$ the points have pairwise distinct x-coordinates. We further write $b^{(h)} = \max\{b_k \mid 1 \leqslant i \leqslant p \text{ and } (x_k, y_k) \in \mathcal{P}_h\}$ for each h, and $\beta = \sum_{1\leqslant h\leqslant q} b^{(h)}$. Then, the cost bound is $\mathcal{O}((m+\beta)^{\omega-1}\mathsf{M}(\sigma)\log(\sigma)^2)$, with a probabilistic algorithm (Chowdhury et al., 2015, Section IV.C). We note that β depends on the chosen partition of the points. The algorithm in this paper is deterministic and has a better cost.

All the mentioned algorithms for the interpolation step of list- or soft-decoding of Reed–Solomon codes, including the one presented in this paper, can be used in conjunction with the *re-encoding technique* (Welch and Berlekamp, 1986; Kötter and Vardy, 2003b) for the decoding problem.

2.7. Applications of the multivariate case

The case $r > 1$ is used for example in the interpolation steps of the list-decoding of Parvaresh–Vardy codes (Parvaresh and Vardy, 2005) and of folded Reed–Solomon codes (Guruswami and Rudra, 2008), as well as in Private Information Retrieval (Devet et al., 2012). In these contexts, one deals with Problem 4 for some $r > 1$ with, in most cases, $\Gamma = \{j \in \mathbb{N}^r \mid |j| < a\}$ where a is the list-size parameter, and $\mu_k = \{(i, j) \in \mathbb{N} \times \mathbb{N}^r \mid i + |j| < b\}$ where b is the multiplicity parameter. Then, $m = \binom{r+a}{r}$ and $\sigma = \binom{r+b}{r+1} p$. Besides, the weight (as mentioned in Subsection 2.4) is $\mathbf{w} = (w, \ldots, w) \in \mathbb{N}^r$ for a fixed positive integer w; then, the corresponding input shift is $\mathbf{s} = (|j|w)_{j\in\Gamma}$.

To our knowledge, the best known cost has been obtained by a probabilistic algorithm which uses $\mathcal{O}(m^{\omega-1}\mathsf{M}(\sigma)\log(\sigma)^2)$ operations (Chowdhury et al., 2015, Theorem 1) to compute one solution with some degree constraints related to $\mathbf{s}$. However, this is not satisfactory for the application to Private Information Retrieval, where one wants an $\mathbf{s}$-minimal basis of solutions: using the polynomial matrix reduction approach mentioned in Subsections 2.5 and 2.6, such a basis can be computed in $\mathcal{O}(m^\omega \mathsf{M}(bp)(\log(m)^2 + \log(bp)))$ operations (Busse, 2008; Brander, 2010; Cohn and Heninger, 2012–2013) (we note that $mbp > \sigma$). It is not clear to us what bound we have on $|\mathbf{s}|$ in this context, and thus how our algorithm compares to these two results.

3. A divide-and-conquer algorithm

In this section, we assume that $\mathbf{J} \in \mathbb{K}^{\sigma\times\sigma}$ is a Jordan matrix given by means of a standard representation as in (5), and we provide a description of our divide-and-conquer algorithm together with a proof of the following refinement of our main result, Theorem 1.5.

Proposition 3.1. *Without loss of generality, assume that* $\min(\mathbf{s}) = 0$*; then, let* $\xi = |\mathbf{s}|$*. Algorithm 1 solves Problem 1 deterministically, and the sum of the* $\mathbf{s}$*-row degrees of the computed* $\mathbf{s}$*-minimal interpolation basis is at most* $\sigma + \xi$*. If* $\sigma > m$*, a cost bound for this algorithm is given by*

$$\mathcal{O}(m^{\omega-1}(\mathsf{M}(\sigma) + \sigma\log(m)) + m^{\omega}\mathsf{M}(\sigma/m)\log(\sigma/m)^2 + m\mathsf{M}(\sigma)\log(\sigma)\log(\sigma/m) + m^{\omega-1}\mathsf{M}(\xi) + m^{\omega}\mathsf{M}(\xi/m)\log(\xi/m)) \quad \text{if } \omega > 2,$$

$$\mathcal{O}(m\mathsf{M}(\sigma)(\log(m)^3 + \log(\sigma)\log(\sigma/m)) + m^2\mathsf{M}(\sigma/m)\log(\sigma)\log(\sigma/m)\log(m) + m\mathsf{M}(\xi)\log(m)^2 + m^2\mathsf{M}(\xi/m)\log(\xi/m)\log(m)) \quad \text{if } \omega = 2;$$

if $\sigma \leqslant m$*, it is given in Proposition 7.1.*

This algorithm relies on some subroutines, for which cost estimates are given in the following sections and are taken for granted here:

- in Section 4, the fast multiplication of two polynomial matrices with respect to the average row degree of the operands and of the result;
- in Section 5, the *change of shift*: given an $\mathbf{s}$-minimal interpolation basis $\mathbf{P}$ and some shift $\mathbf{t}$, compute an $(\mathbf{s}+\mathbf{t})$-minimal interpolation basis;
- in Section 6, the fast computation of a product of the form $\mathbf{P}\cdot\mathbf{E}$;
- in Section 7, the computation of an $\mathbf{s}$-minimal interpolation basis using linear algebra, which is used here for the base case of the recursion.

First, we focus on the divide-and-conquer subroutine given in Algorithm 2. In what follows, $\mathbf{J}^{(1)}$ and $\mathbf{J}^{(2)}$ always denote the leading and trailing principal $\sigma/2 \times \sigma/2$ submatrices of $\mathbf{J}$. These two submatrices are still in Jordan canonical form, albeit not necessarily in standard representation; this can however be restored by a single pass through the array.

Lemma 3.2. *Algorithm 2 solves Problem 1 deterministically for the uniform shift* $\mathbf{s} = \mathbf{0}$*, and the sum of the row degrees of the computed* $\mathbf{0}$*-minimal interpolation basis is at most* σ*. If* $\sigma > m$*, a cost bound for this algorithm is given by*

$$\mathcal{O}(m^{\omega-1}(\mathsf{M}(\sigma) + \sigma\log(m)) + m^{\omega}\mathsf{M}(\sigma/m)\log(\sigma/m)^2 + m\mathsf{M}(\sigma)\log(\sigma)\log(\sigma/m)) \quad \text{if } \omega > 2,$$

$$\mathcal{O}(m\mathsf{M}(\sigma)(\log(m)^3 + \log(\sigma)\log(\sigma/m)) + m^2\mathsf{M}(\sigma/m)\log(\sigma)\log(\sigma/m)\log(m)) \quad \text{if } \omega = 2;$$

if $\sigma \leqslant m$*, it is given in Proposition 7.1.*

In the rest of this paper, we use convenient notation for the cost of polynomial matrix multiplication and for related quantities that arise when working with submatrices of a given degree range as well as in divide-and-conquer computations. Hereafter, $\log(\cdot)$ always stands for the logarithm in base 2.

Definition 3.3. Let m and d be two positive integers. Then, $\mathsf{MM}(m,d)$ is such that two matrices in $\mathbb{K}[X]^{m\times m}$ of degree at most d can be multiplied using $\mathsf{MM}(m,d)$ operations in $\mathbb{K}$. Then, writing $\bar{m}$ and $\bar{d}$ for the smallest powers of 2 at least m and d, we also define

- $\mathsf{MM}'(m,d) = \sum_{0\leqslant i\leqslant\log(\bar{m})} 2^i\mathsf{MM}(2^{-i}\bar{m}, 2^i\bar{d})$,
- $\overline{\mathsf{MM}'}(m,d) = \sum_{0\leqslant i\leqslant\log(\bar{m})} 2^i\mathsf{MM}'(2^{-i}\bar{m}, 2^i\bar{d})$,
- $\mathsf{MM}''(m,d) = \sum_{0\leqslant i\leqslant\log(\bar{d})} 2^i\mathsf{MM}(\bar{m}, 2^{-i}\bar{d})$,
- $\overline{\mathsf{MM}''}(m,d) = \sum_{0\leqslant i\leqslant\log(\bar{m})} 2^i\mathsf{MM}''(2^{-i}\bar{m}, 2^i\bar{d})$.

We note that one can always take $\mathsf{MM}(m,d) \in \mathcal{O}(m^{\omega}\mathsf{M}(d))$. Upper bounds for the other quantities are detailed in Appendix A.

Algorithm 1 *(*InterpolationBasis*).*

Input:

- a matrix $\mathbf{E} \in \mathbb{K}^{m\times\sigma}$,
- a Jordan matrix $\mathbf{J} \in \mathbb{K}^{\sigma\times\sigma}$ in standard representation,
- a shift $\mathbf{s} \in \mathbb{N}^m$.

Output: an $\mathbf{s}$-minimal interpolation basis $\mathbf{P} \in \mathbb{K}[X]^{m\times m}$ for $(\mathbf{E}, \mathbf{J})$.

1. If $\sigma \leqslant m$, Return LinearizationInterpolationBasis$(\mathbf{E}, \mathbf{J}, \mathbf{s}, 2^{\lceil \log(\sigma) \rceil})$
2. Else
  a. $\mathbf{P} \leftarrow$ InterpolationBasisRec$(\mathbf{J}, \mathbf{E})$
  b. Return Shift$(\mathbf{P}, \mathbf{0}, \mathbf{s})$

Algorithm 2 *(*InterpolationBasisRec*).*

Input:

- a matrix $\mathbf{E} \in \mathbb{K}^{m\times\sigma}$,
- a Jordan matrix $\mathbf{J} \in \mathbb{K}^{\sigma\times\sigma}$ in standard representation.

Output: a $\mathbf{0}$-minimal interpolation basis $\mathbf{P} \in \mathbb{K}[X]^{m\times m}$ for $(\mathbf{E}, \mathbf{J})$.

1. If $\sigma \leqslant m$, Return LinearizationInterpolationBasis$(\mathbf{E}, \mathbf{J}, \mathbf{0}, 2^{\lceil \log(\sigma) \rceil})$
2. Else
  a. $\mathbf{E}^{(1)} \leftarrow$ first $\lfloor \sigma/2 \rfloor$ columns of $\mathbf{E}$
  b. $\mathbf{P}^{(1)} \leftarrow$ InterpolationBasisRec$(\mathbf{E}^{(1)}, \mathbf{J}^{(1)})$
  c. $\mathbf{E}^{(2)} \leftarrow$ last $\lceil \sigma/2 \rceil$ columns of $\mathbf{P}^{(1)} \cdot \mathbf{E} =$ ComputeResiduals$(\mathbf{J}, \mathbf{P}^{(1)}, \mathbf{E})$
  d. $\mathbf{P}^{(2)} \leftarrow$ InterpolationBasisRec$(\mathbf{E}^{(2)}, \mathbf{J}^{(2)})$
  e. $\mathbf{R}^{(2)} \leftarrow$ Shift$(\mathbf{P}^{(2)}, \mathbf{0}, \mathrm{rdeg}(\mathbf{P}^{(1)}))$
  f. Return UnbalancedMultiplication$(\mathbf{P}^{(1)}, \mathbf{R}^{(2)}, \sigma)$

Proof of Lemma 3.2. Concerning the base case $\sigma \leqslant m$, the correctness and the cost bound both follow from results that will be given in Section 7, in particular Proposition 7.1.

Let us now consider the case of $\sigma > m$, and assume that $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$, as computed by the recursive calls, are $\mathbf{0}$-minimal interpolation bases for $(\mathbf{E}^{(1)}, \mathbf{J}^{(1)})$ and $(\mathbf{E}^{(2)}, \mathbf{J}^{(2)})$, respectively. The input $\mathbf{E}$ takes the form $\mathbf{E} = [\mathbf{E}^{(1)} | *]$ and we have $\mathbf{P}^{(1)} \cdot \mathbf{E} = [\,\mathbf{0}\,|\mathbf{E}^{(2)}]$ as well as $\mathbf{P}^{(2)} \cdot \mathbf{E}^{(2)} = 0$. Besides, $\mathbf{R}^{(2)}$ is by construction unimodularly equivalent to $\mathbf{P}^{(2)}$, so there exists $\mathbf{U}$ unimodular such that $\mathbf{R}^{(2)} = \mathbf{U}\mathbf{P}^{(2)}$. Defining $\mathbf{P} = \mathbf{R}^{(2)}\mathbf{P}^{(1)}$, our goal is to show that $\mathbf{P}$ is a minimal interpolation basis for $(\mathbf{E}, \mathbf{J})$.

First, since $\mathbf{J}$ is upper triangular we have $\mathbf{P} \cdot \mathbf{E} = \mathbf{R}^{(2)} \cdot [\,\mathbf{0}\,|\mathbf{E}^{(2)}] = [\,\mathbf{0}\,|\mathbf{U}\mathbf{P}^{(2)} \cdot \mathbf{E}^{(2)}] = 0$, so that every row of $\mathbf{P}$ is an interpolant for $(\mathbf{E}, \mathbf{J})$. Let us now consider an arbitrary interpolant $\mathbf{p} \in \mathbb{K}[X]^{1\times m}$ for $(\mathbf{E}, \mathbf{J})$. Then, $\mathbf{p}$ is in particular an interpolant for $(\mathbf{E}^{(1)}, \mathbf{J}^{(1)})$: there exists some row vector $\mathbf{V}$ such that $\mathbf{p} = \mathbf{V}\mathbf{P}^{(1)}$. Furthermore, the equalities $0 = \mathbf{p} \cdot \mathbf{E} = \mathbf{V}\mathbf{P}^{(1)} \cdot \mathbf{E} = [\,\mathbf{0}\,|\mathbf{v} \cdot \mathbf{E}^{(2)}]$ show that $\mathbf{v} \cdot \mathbf{E}^{(2)} = 0$. Thus, there exists some row vector $\mathbf{w}$ such that $\mathbf{v} = \mathbf{w}\mathbf{P}^{(2)}$, which gives $\mathbf{p} = \mathbf{w}\mathbf{P}^{(2)}\mathbf{P}^{(1)} = \mathbf{w}\mathbf{U}^{-1}\mathbf{P}$. This means that every interpolant for $(\mathbf{E}, \mathbf{J})$ is a $\mathbb{K}[X]$-linear combination of the rows of $\mathbf{P}$.

Then, it remains to check that $\mathbf{P}$ is $\mathbf{0}$-reduced. As a $\mathbf{0}$-minimal interpolation basis, $\mathbf{P}^{(2)}$ is $\mathbf{0}$-reduced and has full rank. Then, the construction of $\mathbf{R}^{(2)}$ using Algorithm Shift ensures that it is $\mathbf{t}$-reduced, where $\mathbf{t} = \mathrm{rdeg}(\mathbf{P}^{(1)})$. Define $\mathbf{u} = \mathrm{rdeg}(\mathbf{P}) = \mathrm{rdeg}(\mathbf{R}^{(2)}\mathbf{P}^{(1)})$; the predictable-degree property (Kailath, 1980, Theorem 6.3-13) implies that $\mathbf{u} = \mathrm{rdeg}_{\mathbf{t}}(\mathbf{R}^{(2)})$. Using the identity $\mathbf{X}^{-\mathbf{u}}\mathbf{P} = \mathbf{X}^{-\mathbf{u}}\mathbf{R}^{(2)}\mathbf{X}^{\mathbf{t}}\mathbf{X}^{-\mathbf{t}}\mathbf{P}^{(1)}$, we obtain that the $\mathbf{0}$-leading matrix of $\mathbf{P}$ is the product of the $\mathbf{t}$-leading matrix of $\mathbf{R}^{(2)}$ and the $\mathbf{0}$-leading matrix of $\mathbf{P}^{(1)}$, which are both invertible. Thus, the $\mathbf{0}$-leading matrix of $\mathbf{P}$ is invertible as well, and therefore $\mathbf{P}$ is $\mathbf{0}$-reduced.

Thus, for any σ, the algorithm correctly computes an $\mathbf{s}$-minimal interpolation basis for $(\mathbf{E}, \mathbf{J})$. As shown in Section 1, there is a direct link between M-Padé approximation and shifted minimal interpolation bases with a multiplication matrix in Jordan canonical form. Then, the result in Van Barel and Bultheel (1992, Theorem 4.1) proves that for a given σ, the determinant of a $\mathbf{0}$-minimal interpolation basis $\mathbf{P}$ has degree at most σ. Hence $|\mathrm{rdeg}(\mathbf{P})| = \deg(\det(\mathbf{P})) \leqslant \sigma$ follows the fact that the sum of the row degrees of a $\mathbf{0}$-reduced matrix equals the degree of its determinant.

Let us finally prove the announced cost bound for $\sigma > m$. Without loss of generality, we assume that σ/m is a power of 2. Each of Steps **2.b** and **2.d** calls the algorithm recursively on an instance of Problem 1 with dimensions m and $\sigma/2$.

- The leaves of the recursion are for $m/2 \leqslant \sigma \leqslant m$ and thus according to Proposition 7.1 each of them uses $\mathcal{O}(m^\omega \log(m))$ operations if $\omega > 2$, and $\mathcal{O}(m^2 \log(m)^2)$ operations if $\omega = 2$. For $\sigma > m$ (with σ/m a power of 2), the recursion leads to σ/m leaves, which thus yield a total cost of $\mathcal{O}(m^{\omega-1}\sigma \log(m))$ operations if $\omega > 2$, and $\mathcal{O}(m\sigma \log(m)^2)$ operations if $\omega = 2$.
- According to Proposition 6.1, Step **2.c** uses $\mathcal{O}(\mathsf{MM}(m, \sigma/m)\log(\sigma/m) + m\mathsf{M}(\sigma)\log(\sigma))$ operations. Using the super-linearity property of $d \mapsto \mathsf{MM}(m, d)$, we see that this contributes to the total cost as $\mathcal{O}(\mathsf{MM}(m, \sigma/m)\log(\sigma/m)^2 + m\mathsf{M}(\sigma)\log(\sigma)\log(\sigma/m))$ operations.
- For Step **2.e**, we use Proposition 5.2 with $\xi = \sigma$, remarking that both the sum of the entries of $\mathbf{t} = \mathrm{rdeg}(\mathbf{P}^{(1)})$ and that of $\mathrm{rdeg}(\mathbf{P}^{(2)})$ are at most $\sigma/2$. Then, the change of shift is performed using $\mathcal{O}(\overline{\mathsf{MM}'}(m, \sigma/m) + \overline{\mathsf{MM}''}(m, \sigma/m))$ operations. Thus, altogether the time spent in this step is $\mathcal{O}(\sum_{0\leqslant i\leqslant \log(\sigma/m)} 2^i(\overline{\mathsf{MM}'}(m, 2^{-i}\sigma/m) + \overline{\mathsf{MM}''}(m, 2^{-i}\sigma/m)))$ operations; we give an upper bound for this quantity in Lemma A.3.
- From Proposition 5.2 we obtain that $\mathrm{rdeg}_{\mathbf{t}}(\mathbf{R}^{(2)}) \leqslant |\mathrm{rdeg}(\mathbf{P}^{(2)})| + |\mathbf{t}| \leqslant \sigma$. Then, using Proposition 4.1 with $\xi = \sigma$, the polynomial matrix multiplication in Step **2.f** can be done in time $\mathcal{O}(\mathsf{MM}'(m, \sigma/m))$. Besides, it is easily verified that by definition $\mathsf{MM}'(m, \sigma/m) \leqslant \overline{\mathsf{MM}'}(m, \sigma/m)$, so that the cost for this step is dominated by the cost for the change of shift.

Adding these costs and using the bounds in Appendix A leads to the conclusion. □

We now prove our main result.

Proof of Proposition 3.1. The correctness of Algorithm 1 follows from the correctness of Algorithms 2, 4 and 9. Concerning the cost bound when $\sigma > m$, Lemma 3.2 gives the number of operations used by Step **2.a** to produce $\mathbf{P}$, which satisfies $|\mathrm{rdeg}(\mathbf{P})| \leqslant \sigma$. Then, considering $\min(\mathbf{s}) = 0$ without loss of generality, we have $|\mathrm{rdeg}(\mathbf{P})| + |\mathbf{s}| \leqslant \sigma + \xi$: Proposition 5.2 states that Step **2.b** can be performed using $\mathcal{O}(\overline{\mathsf{MM}'}(m, (\sigma+\xi)/m) + \overline{\mathsf{MM}''}(m, (\sigma+\xi)/m))$ operations. The cost bound then follows from the bounds in Lemma A.2. Furthermore, from Proposition 5.2 we also know that the sum of the $\mathbf{s}$-row degrees of the output matrix is exactly $|\mathrm{rdeg}(\mathbf{P})| + |\mathbf{s}|$, which is itself at most $\sigma + \xi$. □

4. Multiplying matrices with unbalanced row degrees

In this section, we give a detailed complexity analysis concerning the fast algorithm from Zhou et al. (2012, Section 3.6) for the multiplication of matrices with controlled, yet possibly unbalanced, row degrees. For completeness, we recall this algorithm below in Algorithm 3. It is a central building block in our algorithm: it is used in the multiplication of interpolation bases (Step **2.f** of Algorithm 2) and also for the multiplication of nullspace bases that occur in the algorithm of Zhou et al. (2012), which we use to perform the change of shift in Step **2.e** of Algorithm 2.

In the rest of the paper, most polynomial matrices are interpolation bases and thus are square and nonsingular. In contrast, in this section polynomial matrices may be rectangular and have zero rows, since this may occur in nullspace basis computations (see Appendix B). Thus, we extend the definitions of shift and row degree as follows. For a matrix $\mathbf{A} = [a_{i,j}]_{i,j}$ in $\mathbb{K}[X]^{k\times m}$ for some k, and a shift $\mathbf{s} = (s_1, \ldots, s_m) \in (\mathbb{N} \cup \{-\infty\})^m$, the $\mathbf{s}$-row degree of $\mathbf{A}$ is the tuple $\mathrm{rdeg}_{\mathbf{s}}(\mathbf{A}) = (d_1, \ldots, d_k) \in (\mathbb{N} \cup \{-\infty\})^k$, with $d_i = \max_j(\deg(a_{i,j}) + s_j)$ for all i, and using the usual convention that $\deg(0) = -\infty$. Besides, $|\mathbf{s}|$ denotes the sum of the non-negative entries of $\mathbf{s}$, that is, $|\mathbf{s}| = \sum_{1\leqslant i\leqslant k, s_i \neq -\infty} s_i$.

In order to multiply matrices with unbalanced row degrees, we use in particular a technique based on *partial linearization*, which can be seen as a simplified version of the one in Gupta et al. (2012, Section 6) for the purpose of multiplication. For a matrix $\mathbf{B}$ with sum of row degrees ξ, meant to be the left operand in a product $\mathbf{BA}$ for some $\mathbf{A} \in \mathbb{K}[X]^{m\times m}$, this technique consists in expanding the high-degree rows of $\mathbf{B}$ so as to obtain a matrix $\widetilde{\mathbf{B}}$ with $\mathcal{O}(m)$ rows and degree at most ξ/m, then computing the product $\widetilde{\mathbf{B}}\mathbf{A}$, and finally retrieving the actual product $\mathbf{BA}$ by grouping together the rows that have been expanded (called *partial compression* in what follows).

More precisely, let $\mathbf{B} \in \mathbb{K}[X]^{k\times m}$ for some k and m, with $\mathrm{rdeg}(\mathbf{B}) = (d_1, \ldots, d_k)$ and write $\xi = d_1 + \cdots + d_k$. We are given a target degree bound d. For each i, the row $\mathbf{B}_{i,*}$ of degree d_i is expanded into $\alpha_i = 1 + \lfloor d_i/(d+1)\rfloor$ rows $\widetilde{\mathbf{B}}_{(i,0),*}, \ldots, \widetilde{\mathbf{B}}_{(i,\alpha_i-1),*}$ of degree at most d, related by the identity

$$\mathbf{B}_{i,*} = \widetilde{\mathbf{B}}_{(i,0),*} + X^{d+1}\widetilde{\mathbf{B}}_{(i,1),*} + \cdots + X^{(\alpha_i-1)(d+1)}\widetilde{\mathbf{B}}_{(i,\alpha_i-1),*}. \tag{9}$$

Then, the expanded matrix $\widetilde{\mathbf{B}}$ has $\sum_{1\leqslant i\leqslant k}\alpha_i \leqslant k + \xi/(d+1)$ rows $\widetilde{\mathbf{B}}_{(i,j),*}$. We will mainly use this technique for $k \leqslant m$ and $d = \lfloor \xi/m\rfloor$ or $d = \lceil \xi/m\rceil$, in which case $\widetilde{\mathbf{B}}$ has fewer than $2m$ rows. The partial compression is the computation of the row i of the product $\mathbf{BA}$ from the rows $(i,0), \ldots, (i,\alpha_i-1)$ of $\widetilde{\mathbf{B}}\mathbf{A}$ using the formula in (9).

Proposition 4.1. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *in* $\mathbb{K}[X]^{m\times m}$*, and* $\mathbf{d} = \mathrm{rdeg}(\mathbf{A})$*. Let* $\xi \geqslant m$ *be an integer such that* $|\mathbf{d}| \leqslant \xi$ *and* $|\mathrm{rdeg}_{\mathbf{d}}(\mathbf{B})| \leqslant \xi$*. Then, the product* $\mathbf{BA}$ *can be computed using*

$$\begin{aligned} &\mathcal{O}(\mathsf{MM}'(m, \xi/m)) \\ &\quad \subseteq \mathcal{O}(m^{\omega-1}\mathsf{M}(\xi)) && \text{if } \omega > 2 \\ &\quad \subseteq \mathcal{O}(m\mathsf{M}(\xi)\log(m)) && \text{if } \omega = 2 \end{aligned}$$

operations in $\mathbb{K}$.

Proof. In this proof, we use notation from Algorithm 3. The correctness of this algorithm follows from the identity $\mathbf{BA} = \hat{\mathbf{B}}\hat{\mathbf{A}} = \mathbf{B}_0\mathbf{A}_0 + \mathbf{B}_1\mathbf{A}_1 + \cdots + \mathbf{B}_\ell\mathbf{A}_\ell$. In what follows we focus on proving the cost bound $\mathcal{O}(\mathsf{MM}'(m, \xi/m))$; the announced upper bounds on this quantity follow from Lemma A.1.

We start with Step **6**, which adds the $m \times m$ matrices $\mathbf{P}_i = \mathbf{B}_i\mathbf{A}_i$ obtained after the first five steps. For each i in $\{0, \ldots, \ell\}$, we have $\mathrm{rdeg}(\mathbf{P}_i) \leqslant \mathrm{rdeg}(\mathbf{BA}) \leqslant \mathrm{rdeg}_{\mathbf{d}}(\mathbf{B})$ componentwise, hence $|\mathrm{rdeg}(\mathbf{P}_i)| \leqslant \xi$. Recalling that $\ell = \lceil \log(m)\rceil$, the sum at Step **6** thus uses $\mathcal{O}(m\xi\ell) = \mathcal{O}(m\xi\log(m))$ additions in $\mathbb{K}$. On the other hand, by definition of $\mathsf{MM}'(\cdot,\cdot)$, the trivial lower bound $\mathsf{MM}(n, d) \geqslant n^2 d$ for any n, d implies that $m\xi\log(m) \in \mathcal{O}(\mathsf{MM}'(m, \xi/m))$.

Now we study the for loop. We remark that only Step **5.c** involves arithmetic operations in $\mathbb{K}$. Therefore the main task is to give bounds on the dimensions and degrees of the matrices we multiply at Step **5.c**. For i in $\{0, \ldots, \ell\}$, the column dimension of $\mathbf{A}_i$ is m and the row dimension of $\mathbf{B}_i^{\emptyset}$ is k_i. We further denote by m_i the row dimension of $\mathbf{A}_i$ (and column dimension of $\widetilde{\mathbf{B}}_i^{\emptyset}$), and we write $[d_{\pi(1)}, \ldots, d_{\pi(m)}] = [\mathbf{d}_0|\cdots|\mathbf{d}_\ell]$ where the sizes of $\mathbf{d}_0, \ldots, \mathbf{d}_\ell$ correspond to those of the blocks of $\hat{\mathbf{B}}$ as in Step **4** of the algorithm.

First, let $i = 0$. Then, $\mathbf{A}_0$ is $m_0 \times m$ of degree at most ξ/m and $\mathbf{B}_0$ is $k_0 \times m_0$ with $m_0 \leqslant m$ and $k_0 \leqslant m$ (we note that for $i = 0$ these may be equalities and thus one does not need to discard the zero rows of $\mathbf{B}_0$ to obtain efficiency). Besides, we have the componentwise inequality $\mathrm{rdeg}(\mathbf{B}_0) \leqslant \mathrm{rdeg}_{\mathbf{d}_0}(\mathbf{B}_0) \leqslant \mathrm{rdeg}_{\mathbf{d}}(\mathbf{B})$, so that $|\mathrm{rdeg}(\mathbf{B}_0)| \leqslant |\mathrm{rdeg}_{\mathbf{d}}(\mathbf{B})| \leqslant \xi$. Then, $\mathbf{B}_0^{\emptyset}$ can be partially linearized into a matrix $\widetilde{\mathbf{B}}_0^{\emptyset}$ which has at most $2m$ rows and degree at most ξ/m, and the computation at Step **5.c** for $i = 0$ uses $\mathcal{O}(\mathsf{MM}(m, \xi/m))$ operations.

Now, let $i \in \{1, \ldots, \ell\}$. By assumption, the sum of the row degrees of $\mathbf{A}$ does not exceed ξ: since all rows in $\mathbf{A}_i$ have degree more than $2^{i-1}\xi/m$, this implies that $m_i < m/2^{i-1}$. Besides, since $\min(\mathbf{d}_i) > 2^{i-1}\xi/m$, we obtain that every nonzero row of $\mathbf{B}_i$ has $\mathbf{d}_i$-row degree more than $2^{i-1}\xi/m$. Then, $\xi \geqslant |\mathrm{rdeg}_{\mathbf{d}}(\mathbf{B})| \geqslant |\mathrm{rdeg}_{\mathbf{d}_i}(\mathbf{B}_i^{\emptyset})| > k_i 2^{i-1}\xi/m$ implies that $k_i < m/2^{i-1}$. Furthermore, since we have $|\mathrm{rdeg}(\mathbf{B}_i^{\emptyset})| = |\mathrm{rdeg}(\mathbf{B}_i)| \leqslant \xi$, the partial linearization at Step **5.b** can be done by at most doubling the number of rows of $\mathbf{B}_i^{\emptyset}$, producing $\widetilde{\mathbf{B}}_i^{\emptyset}$ with fewer than $2m/2^{i-1}$ rows and of degree at most $2^{i-1}\xi/m$. To summarize: $\mathbf{A}_i$ has m columns, $m_i < m/2^{i-1}$ rows, and degree at most $2^i\xi/m$; $\widetilde{\mathbf{B}}_i^{\emptyset}$ has fewer than $2m/2^{i-1}$ rows, and degree less than $2^i\xi/m$. Then, the computation of $\widetilde{\mathbf{P}}_i^{\emptyset}$ uses $\mathcal{O}(2^i\mathsf{MM}(m/2^{i-1}, 2^i\xi/m))$ operations in $\mathbb{K}$. Thus, overall the for loop uses $\mathcal{O}(\mathsf{MM}'(m, \xi/m))$ operations in $\mathbb{K}$. $\square$

Algorithm 3 (UNBALANCEDMULTIPLICATION).

Input:

- polynomial matrices $\mathbf{A}$ and $\mathbf{B}$ in $\mathbb{K}[X]^{m\times m}$,
- an integer ξ with $\xi \geqslant m$, $|\mathbf{d}| \leqslant \xi$, and $|\mathrm{rdeg}_{\mathbf{d}}(\mathbf{B})| \leqslant \xi$ where $\mathbf{d} = \mathrm{rdeg}(\mathbf{A})$.

Output: the product $\mathbf{P} = \mathbf{BA}$.

1. $\pi \leftarrow$ a permutation of $\{1, \dots, m\}$ such that $(d_{\pi(1)}, \dots, d_{\pi(m)})$ is non-decreasing
2. $\hat{\mathbf{A}} \leftarrow \pi\mathbf{A}$ and $\hat{\mathbf{B}} \leftarrow \mathbf{B}\pi^{-1}$
3. define $\ell = \lceil \log(m) \rceil$ and the row blocks $\hat{\mathbf{A}} = [\mathbf{A}_0^{\mathsf{T}}|\mathbf{A}_1^{\mathsf{T}}|\cdots|\mathbf{A}_\ell^{\mathsf{T}}]^{\mathsf{T}}$, where the rows in $\mathbf{A}_0$ have row degree at most ξ/m and for $i = 1, \dots, \ell$ the rows in $\mathbf{A}_i$ have row degree in $\{2^{i-1}\xi/m + 1, \dots, 2^i\xi/m\}$
4. define $\hat{\mathbf{B}} = [\mathbf{B}_0|\mathbf{B}_1|\cdots|\mathbf{B}_\ell]$ the corresponding column blocks of $\hat{\mathbf{B}}$
5. For i from 0 to ℓ:
  a. Read $r_1, \dots, r_{k_i}$ the indices of the nonzero rows in $\mathbf{B}_i$ and define $\mathbf{B}_i^{\emptyset}$ the submatrix of $\mathbf{B}_i$ obtained by removing the zero rows
  b. $\widetilde{\mathbf{B}}_i^{\emptyset} \leftarrow$ partial linearization of $\mathbf{B}_i^{\emptyset}$ with $\deg(\widetilde{\mathbf{B}}_i^{\emptyset}) \leqslant 2^i\xi/m$
  c. $\widetilde{\mathbf{P}}_i^{\emptyset} \leftarrow \widetilde{\mathbf{B}}_i^{\emptyset}\mathbf{A}_i$
  d. Perform the partial compression $\mathbf{P}_i^{\emptyset} \leftarrow \widetilde{\mathbf{P}}_i^{\emptyset}$
  e. Re-introduce the zero rows to obtain $\mathbf{P}_i$, which is $\mathbf{B}_i\mathbf{A}_i$ (its rows at indices $r_1, \dots, r_{k_i}$ are those of $\mathbf{P}_i^{\emptyset}$, its other rows are zero)
6. Return $\mathbf{P} = \mathbf{P}_0 + \mathbf{P}_1 + \cdots + \mathbf{P}_\ell$

5. Fast shifted reduction of a reduced matrix

In Algorithm 1, a key ingredient to achieve efficiency is to control the size of the intermediate interpolation bases that are computed in recursive calls. For this, we compute all minimal bases for the *uniform* shift and then recover the *shifted* minimal basis using what we call a change of shift, that we detail in this section. More precisely, we are interested in the ability to transform an $\mathbf{s}$-reduced matrix $\mathbf{P} \in \mathbb{K}[X]^{m\times m}$ with full rank into a unimodularly equivalent matrix that is $(\mathbf{s}+\mathbf{t})$-reduced for some given shift $\mathbf{t} \in \mathbb{N}^m$: this is the problem of polynomial lattice reduction for the shift $\mathbf{s}+\mathbf{t}$, knowing that the input matrix is already reduced for the shift $\mathbf{s}$.

Compared to a general row reduction algorithm such as the one in Gupta et al. (2012), our algorithm achieves efficient computation with regards to the average row degree of the input $\mathbf{P}$ rather than the maximum degree of the entries of $\mathbf{P}$. The main consequence of having an $\mathbf{s}$-*reduced* input $\mathbf{P}$ is that no high-degree cancellation can occur when performing unimodular transformations on the rows of $\mathbf{P}$, which is formalized as the predictable-degree property (Kailath, 1980, Theorem 6.3-13). In particular, the unimodular transformation between $\mathbf{P}$ and an $(\mathbf{s}+\mathbf{t})$-reduced equivalent matrix has small row degree, and the proposition below shows how to exploit this to solve our problem via the computation of a shifted minimal (left) nullspace basis of some $2m \times m$ polynomial matrix. We remark that similar ideas about the use of minimal nullspace bases to compute reduced forms were already in Beelen et al. (1988, Section 3).

Lemma 5.1. *Let $\mathbf{s} \in \mathbb{N}^m$ and $\mathbf{t} \in \mathbb{N}^m$, let $\mathbf{P} \in \mathbb{K}[X]^{m\times m}$ be $\mathbf{s}$-reduced and nonsingular, and define $\mathbf{d} = \mathrm{rdeg}_{\mathbf{s}}(\mathbf{P})$. Then $\mathbf{R} \in \mathbb{K}[X]^{m\times m}$ is an $(\mathbf{s}+\mathbf{t})$-reduced form of $\mathbf{P}$ with unimodular transformation $\mathbf{U} = \mathbf{RP}^{-1} \in \mathbb{K}[X]^{m\times m}$ if and only if $[\mathbf{U}|\mathbf{RX}^{\mathbf{s}}]$ is a $(\mathbf{d}, \mathbf{t})$-minimal nullspace basis of $[\mathbf{X}^{\mathbf{s}}\mathbf{P}^{\mathsf{T}}| - \mathbf{I}_m]^{\mathsf{T}}$.*

Proof. We first assume that the result holds for the uniform shift $\mathbf{s} = \mathbf{0} \in \mathbb{N}^m$, and we show that the general case $\mathbf{s} \in \mathbb{N}^m$ follows. Indeed, considering the $\mathbf{0}$-reduced matrix $\mathbf{PX}^{\mathbf{s}}$ we have $\mathbf{d} = \mathrm{rdeg}_{\mathbf{s}}(\mathbf{P}) = \mathrm{rdeg}(\mathbf{PX}^{\mathbf{s}})$. Hence $[\mathbf{U}|\mathbf{R}]$ is a $(\mathbf{d}, \mathbf{t})$-minimal nullspace basis of $[\mathbf{X}^{\mathbf{s}}\mathbf{P}^{\mathsf{T}}| - \mathbf{I}_m]^{\mathsf{T}}$ if and only if $\mathbf{R}$ is a $\mathbf{t}$-reduced form of $\mathbf{PX}^{\mathbf{s}}$ with unimodular transformation $\mathbf{U}$ such that $\mathbf{UPX}^{\mathbf{s}} = \mathbf{R}$; that is, if and only if $\mathbf{RX}^{-\mathbf{s}} \in \mathbb{K}[X]^{m\times m}$ is a $(\mathbf{s}+\mathbf{t})$-reduced form of $\mathbf{P}$ with unimodular transformation $\mathbf{U}$ such that $\mathbf{UP} = \mathbf{RX}^{-\mathbf{s}}$.

Let us now prove the proposition for the uniform shift $\mathbf{s} = \mathbf{0}$. First, we assume that $\mathbf{R} \in \mathbb{K}[X]^{m\times m}$ is a $\mathbf{t}$-reduced form of $\mathbf{P}$ with unimodular transformation $\mathbf{U}$. From $\mathbf{UP} = \mathbf{R}$ it follows that the rows of $[\mathbf{U}|\mathbf{R}]$ are in the nullspace of $[\mathbf{P}^\mathsf{T}| - \mathbf{I}_m]^\mathsf{T}$. Writing $[\mathbf{N}|*]$ with $\mathbf{N} \in \mathbb{K}[X]^{m\times m}$ to denote an arbitrary basis of that nullspace, we have $[\mathbf{U}|\mathbf{R}] = \mathbf{V}[\mathbf{N}|*]$ for some $\mathbf{V} \in \mathbb{K}[X]^{m\times m}$ and thus $\mathbf{U} = \mathbf{VN}$. Since $\mathbf{U}$ is unimodular, $\mathbf{V}$ is unimodular too and $[\mathbf{U}|\mathbf{R}]$ is a basis of the nullspace of $[\mathbf{P}^\mathsf{T}| - \mathbf{I}_m]^\mathsf{T}$. It remains to check that $[\mathbf{U}|\mathbf{R}]$ is $(\mathbf{d}, \mathbf{t})$-reduced. Since $\mathbf{P}$ is reduced, we have $\mathrm{rdeg}_\mathbf{d}(\mathbf{U}) = \mathrm{rdeg}(\mathbf{UP}) = \mathrm{rdeg}(\mathbf{R})$ by the predictable-degree property (Kailath, 1980, Theorem 6.3-13) and, using $\mathbf{t} \geqslant \mathbf{0}$, we obtain $\mathrm{rdeg}_\mathbf{d}(\mathbf{U}) \leqslant \mathrm{rdeg}_\mathbf{t}(\mathbf{R})$. Hence $\mathrm{rdeg}_{(\mathbf{d},\mathbf{t})}([\mathbf{U}|\mathbf{R}]) = \mathrm{rdeg}_\mathbf{t}(\mathbf{R})$ and, since $\mathbf{R}$ is $\mathbf{t}$-reduced, this implies that $[\mathbf{U}|\mathbf{R}]$ is $(\mathbf{d}, \mathbf{t})$-reduced.

Now, let $[\mathbf{U}|\mathbf{R}]$ be a $(\mathbf{d}, \mathbf{t})$-minimal nullspace basis of $[\mathbf{P}^\mathsf{T}| - \mathbf{I}_m]^\mathsf{T}$. First, we note that $\mathbf{U}$ satisfies $\mathbf{U} = \mathbf{RP}^{-1}$. It remains to check that $\mathbf{U}$ is unimodular and that $\mathbf{R}$ is $\mathbf{t}$-reduced. To do this, let $\widehat{\mathbf{R}}$ denote an arbitrary $\mathbf{t}$-reduced form of $\mathbf{P}$ and let $\widehat{\mathbf{U}} = \widehat{\mathbf{R}}\mathbf{P}^{-1}$ be the associated unimodular transformation. From the previous paragraph, we know that $[\widehat{\mathbf{U}}|\widehat{\mathbf{R}}]$ is a basis of the nullspace of $[\mathbf{P}^\mathsf{T}| - \mathbf{I}_m]^\mathsf{T}$, and since by definition $[\mathbf{U}|\mathbf{R}]$ is also such a basis, we have $[\mathbf{U}|\mathbf{R}] = \mathbf{W}[\widehat{\mathbf{U}}|\widehat{\mathbf{R}}]$ for some unimodular matrix $\mathbf{W} \in \mathbb{K}[X]^{m\times m}$. In particular, $\mathbf{U} = \mathbf{W}\widehat{\mathbf{U}}$ is unimodular. Furthermore, the two unimodularly equivalent matrices $[\mathbf{U}|\mathbf{R}]$ and $[\widehat{\mathbf{U}}|\widehat{\mathbf{R}}]$ are $(\mathbf{d}, \mathbf{t})$-reduced, so that they share the same shifted row degree up to permutation (see for instance Kailath, 1980, Lemma 6.3-14). Now, from the previous paragraph, we know that $\mathrm{rdeg}_{(\mathbf{d},\mathbf{t})}([\widehat{\mathbf{U}}|\widehat{\mathbf{R}}]) = \mathrm{rdeg}_\mathbf{t}(\widehat{\mathbf{R}})$, and similarly, having $\mathbf{P}$ reduced, $\mathbf{UP} = \mathbf{R}$, and $\mathbf{t} \geqslant \mathbf{0}$ imply that $\mathrm{rdeg}_{(\mathbf{d},\mathbf{t})}([\mathbf{U}|\mathbf{R}]) = \mathrm{rdeg}_\mathbf{t}(\mathbf{R})$. Thus $\mathrm{rdeg}_\mathbf{t}(\mathbf{R})$ and $\mathrm{rdeg}_\mathbf{t}(\widehat{\mathbf{R}})$ are equal up to permutation, and combining this with the fact that $\mathbf{R} = \mathbf{W}\widehat{\mathbf{R}}$ where $\widehat{\mathbf{R}}$ is $\mathbf{t}$-reduced and $\mathbf{W}$ is unimodular, we conclude that $\mathbf{R}$ is $\mathbf{t}$-reduced. □

This leads to Algorithm 4, and in particular such a change of shift can be computed efficiently using the minimal nullspace basis algorithm of Zhou et al. (2012).

Algorithm 4 *(*Shift*).*

Input:

- a matrix $\mathbf{P} \in \mathbb{K}[X]^{m\times m}$ with full rank,
- two shifts $\mathbf{s}, \mathbf{t} \in \mathbb{N}^m$ such that $\mathbf{P}$ is $\mathbf{s}$-reduced.

Output: an $(\mathbf{s} + \mathbf{t})$-reduced form of $\mathbf{P}$.

1. $\mathbf{d} \leftarrow \mathrm{rdeg}_\mathbf{s}(\mathbf{P})$
2. $[\mathbf{U}|\mathbf{R}] \leftarrow$ MinimalNullspaceBasis($[\mathbf{X}^\mathbf{s}\,\mathbf{P}^\mathsf{T}| - \mathbf{I}_m]^\mathsf{T}, (\mathbf{d}, \mathbf{t})$)
3. Return $\mathbf{RX}^{-\mathbf{s}}$

Proposition 5.2. *Let $\mathbf{s} \in \mathbb{N}^m$ and $\mathbf{t} \in \mathbb{N}^m$, let $\mathbf{P} \in \mathbb{K}[X]^{m\times m}$ have full rank and be $\mathbf{s}$-reduced, and define $\mathbf{d} = \mathrm{rdeg}_\mathbf{s}(\mathbf{P})$. We write ξ to denote a parameter such that $\xi \geqslant m$ and $|\mathbf{d}| + |\mathbf{t}| \leqslant \xi$. Then, an $(\mathbf{s} + \mathbf{t})$-reduced form $\mathbf{R} \in \mathbb{K}[X]^{m\times m}$ of $\mathbf{P}$ and the corresponding unimodular transformation $\mathbf{U} = \mathbf{RP}^{-1} \in \mathbb{K}[X]^{m\times m}$ can be computed using*

$$
\begin{aligned}
&\mathcal{O}(\overline{\mathsf{MM}'}(m, \xi/m) + \overline{\mathsf{MM}''}(m, \xi/m)) \\
&\quad \subseteq \mathcal{O}(m^{\omega-1}\mathsf{M}(\xi) + m^{\omega}\mathsf{M}(\xi/m)\log(\xi/m)) && \text{if } \omega > 2 \\
&\quad \subseteq \mathcal{O}(m\mathsf{M}(\xi)\log(m)^2 + m^2\mathsf{M}(\xi/m)\log(\xi/m)\log(m)) && \text{if } \omega = 2
\end{aligned}
$$

operations in $\mathbb{K}$. Besides, we have $|\mathrm{rdeg}_{\mathbf{s}+\mathbf{t}}(\mathbf{R})| = |\mathbf{d}| + |\mathbf{t}|$.

Proof. Write $\mathbf{u} = (\mathbf{d}, \mathbf{t})$ and $\mathbf{M} = [\mathbf{X}^\mathbf{s}\,\mathbf{P}^\mathsf{T}| - \mathbf{I}_m]^\mathsf{T}$. According to Lemma 5.1, Algorithm 4 is correct: it computes $[\mathbf{U}|\mathbf{R}]$ a $\mathbf{u}$-minimal nullspace basis of $\mathbf{M}$, and returns $\mathbf{RX}^{-\mathbf{s}}$ which is an $(\mathbf{s} + \mathbf{t})$-reduced form of $\mathbf{P}$. For a fast solution, the minimal nullspace basis can be computed using (Zhou et al., 2012, Algorithm 1), which we have rewritten in Appendix B (Algorithm 10) along with a detailed cost analysis.

Here, we show that the requirements of this algorithm on its input are fulfilled in our context. Concerning the input matrix, we note that $\mathbf{M}$ has more rows than columns, and $\mathbf{M}$ has full rank since by assumption $\mathbf{P}$ has full rank. Now, considering the requirement on the input shift, first, each element of the shift $\mathbf{u}$ bounds the corresponding row degree of $\mathbf{M}$; and second, the rows of $\mathbf{M}$ can be permuted before the nullspace computation so as to have $\mathbf{u}$ non-decreasing, and then the columns of the obtained nullspace basis can be permuted back to the original order. In details, we first compute $\mathbf{v}$ being the tuple $\mathbf{u}$ sorted in non-decreasing order together with the corresponding permutation matrix $\pi \in \mathbb{K}^{2m\times 2m}$ such that, when $\mathbf{v}$ and $\mathbf{u}$ are seen as column vectors in $\mathbb{N}^{2m\times 1}$, we have $\mathbf{v} = \pi\mathbf{u}$. Now that $\mathbf{v}$ is non-decreasing and bounds the corresponding row degree of $\pi\mathbf{M}$, we compute $\mathbf{N}$ a $\mathbf{v}$-minimal nullspace basis of $\pi\mathbf{M}$ using Algorithm 10, then, $\mathbf{N}\pi$ is a $\mathbf{u}$-minimal nullspace basis of $\mathbf{M}$. Since by assumption $|\mathbf{v}| = |\mathbf{d}| + |\mathbf{t}| \leqslant \xi$, the announced cost bound follows directly from Proposition B.1 in Appendix B.

Finally, we prove the bound on the sum of the $(\mathbf{s}+\mathbf{t})$-row degrees of $\mathbf{R}$. Since $\mathbf{P}$ is $\mathbf{s}$-reduced and $\mathbf{R}$ is $(\mathbf{s}+\mathbf{t})$-reduced, we have $|\mathbf{d}| = \deg(\det(\mathbf{P}\mathbf{X}^{\mathbf{s}}))$ as well as $|\mathrm{rdeg}_{\mathbf{s}+\mathbf{t}}(\mathbf{R})| = \deg(\det(\mathbf{R}\mathbf{X}^{\mathbf{s}+\mathbf{t}}))$ (Kailath, 1980, Section 6.3.2). Then, we have that $|\mathrm{rdeg}_{\mathbf{s}+\mathbf{t}}(\mathbf{R})| = \deg(\det(\mathbf{U}\mathbf{P}\mathbf{X}^{\mathbf{s}+\mathbf{t}})) = \deg(\det(\mathbf{P}\mathbf{X}^{\mathbf{s}})) + |\mathbf{t}| = |\mathbf{d}| + |\mathbf{t}|$, which concludes the proof. □

6. Computing residuals

Let $\mathfrak{E} = \mathbb{K}^{1\times\sigma}$ and $\mathbf{J} \in \mathbb{K}^{\sigma\times\sigma}$ be as in the introduction; in particular, we suppose that $\mathbf{J}$ is a Jordan matrix, given by a standard representation as in (5). Given $\mathbf{E}$ in $\mathfrak{E}^m = \mathbb{K}^{m\times\sigma}$ and a matrix $\mathbf{P}$ in $\mathbb{K}[X]^{m\times m}$, we show how to compute the product $\mathbf{P}\cdot\mathbf{E} \in \mathfrak{E}^m$. We will often call the result *residual*, as this is the role this vector plays in our main algorithm.

To give our complexity estimates, we will make two assumptions, namely that $m \leqslant \sigma$ and that the sum of the row degrees of $\mathbf{P}$ is in $\mathcal{O}(\sigma)$; they will both be satisfied when we apply the following result.

Proposition 6.1. *There exists an algorithm* ComputeResiduals *that computes the matrix $\mathbf{P}\cdot\mathbf{E} \in \mathfrak{E}^m$. If $\sigma \geqslant m$ and if the sum of the row degrees of $\mathbf{P}$ is $\mathcal{O}(\sigma)$, this algorithm uses $\mathcal{O}(\mathsf{MM}(m, \sigma/m)\log(\sigma/m) + m\mathsf{M}(\sigma)\log(\sigma))$ operations in $\mathbb{K}$.*

Remark that when the sum of the row degrees of $\mathbf{P}$ is $\mathcal{O}(\sigma)$, storing $\mathbf{P}$ requires $\mathcal{O}(m\sigma)$ elements in $\mathbb{K}$, so that representing the input and output of this computation involves $\mathcal{O}(m\sigma)$ field elements. At best, one could thus hope for an algorithm of cost $\mathcal{O}(m\sigma)$. Our result is close, as we get a cost of $\mathcal{O}\tilde{~}(m^{1.38}\sigma)$ with the best known value of ω.

6.1. Preliminaries, Chinese remaindering, and related questions

The following lemma writes the output in a more precise manner. The proof is a straightforward consequence of the discussion in Section 1 about writing the notion of interpolant in terms of M-Padé approximation.

Lemma 6.2. *Suppose that $\mathbf{J}$ has the form $((x_1,\sigma_1),\ldots,(x_n,\sigma_n))$. Let $\mathbf{P} \in \mathbb{K}[X]^{m\times m}$ and $\mathbf{E} \in \mathbb{K}^{m\times\sigma}$, and write $\mathbf{E} = [\mathbf{E}_1|\cdots|\mathbf{E}_n]$ with $\mathbf{E}_j$ in $\mathbb{K}^{m\times\sigma_j}$ for $1 \leqslant j \leqslant n$. For $1 \leqslant j \leqslant n$, define the following matrices:*

- $\mathbf{E}_{j,\mathrm{poly}} = \mathbf{E}_j\,[1, X, \ldots, X^{\sigma_j-1}]^\mathsf{T} \in \mathbb{K}[X]^{m\times 1}$ *is the column vector with polynomial entries built from the columns of* $\mathbf{E}_j$,
- $\mathbf{F}_{j,\mathrm{poly}} = \mathbf{P}(X+x_j)\,\mathbf{E}_{j,\mathrm{poly}} \bmod X^{\sigma_j} \in \mathbb{K}[X]^{m\times 1}$,
- $\mathbf{F}_j = [\mathbf{F}_{j,0},\ldots,\mathbf{F}_{j,\sigma_j-1}] \in \mathbb{K}^{m\times\sigma_j}$ *is the matrix whose columns are the coefficients of* $\mathbf{F}_{j,\mathrm{poly}}$ *of degrees* $0,\ldots,\sigma_j-1$.

Then, $\mathbf{P}\cdot\mathbf{E} = \mathbf{F}$ *with* $\mathbf{F} = [\mathbf{F}_1|\cdots|\mathbf{F}_n] \in \mathbb{K}^{m\times\sigma}$.

To give an idea of our algorithm's behaviour, let us first consider the case where $\mathbf{J}$ is the upper shift matrix $\mathbf{Z}$ as in (4), so there is only one Jordan block whose eigenvalue is 0. This corresponds to having $n = 1$ in the previous lemma, which thus says that we can turn the input $\mathbf{E}$ into a vector of m polynomials of degree at most σ, and that we simply have to left-multiply this vector by $\mathbf{P}$. Suppose furthermore that all entries in $\mathbf{P}$ have degree $\mathcal{O}(\sigma/m)$ (this is the most natural situation ensuring that the sum of its row degrees is $\mathcal{O}(\sigma)$, as assumed in Proposition 6.1), so that we have to multiply an $m \times m$ matrix with entries of degree $\mathcal{O}(\sigma/m)$ by an $m \times 1$ vector with entries of degree σ. For this, we use the partial linearization presented in Section 4: we expand the right-hand side into an $m \times m$ polynomial matrix with entries of degree $\mathcal{O}(\sigma/m)$, we multiply it by $\mathbf{P}$, and we recombine the entries of the result; this leads us to the cost $\mathcal{O}(\mathsf{MM}(m, \sigma/m))$.

On the other side of the spectrum, we encountered the case of a diagonal matrix $\mathbf{J}$, with diagonal entries $x_1, \ldots, x_\sigma$ (so all σ_i's are equal to 1); suppose furthermore that these entries are pairwise distinct. In this case, if we let $\mathbf{E}_1, \ldots, \mathbf{E}_\sigma$ be the columns of $\mathbf{E}$, Lemma 6.2 shows that the output is the matrix whose columns are $\mathbf{P}(x_1)\mathbf{E}_1, \ldots, \mathbf{P}(x_\sigma)\mathbf{E}_\sigma$. Evaluating $\mathbf{P}$ at all x_i's would be too costly, as simply representing all the evaluations requires $m^2\sigma$ field elements; instead, we interpolate a column vector of m polynomials $E_1, \ldots, E_m$ of degree less than σ from the respective rows of $\mathbf{E}$, do the same matrix-vector product as above, and evaluate the output at the x_i's; the total cost is $\mathcal{O}(\mathsf{MM}(m, \sigma/m) + m\mathsf{M}(\sigma)\log(\sigma))$.

Our main algorithm generalizes these two particular processes. We now state a few basic results that will be needed for this kind of calculation, around problems related to polynomial modular reduction and Chinese remaindering.

Lemma 6.3. *The following cost estimates hold:*

- *Given p of degree d in $\mathbb{K}[X]$, and x in $\mathbb{K}$, one can compute $p(X + x)$ in $\mathcal{O}(\mathsf{M}(d)\log(d))$ operations in $\mathbb{K}$.*
- *Given moduli $q_1, \ldots, q_s$ in $\mathbb{K}[X]$, whose sum of degrees is e, and p of degree $d + e$, one can compute $p \bmod q_1, \ldots, p \bmod q_s$ using $\mathcal{O}(\mathsf{M}(d) + \mathsf{M}(e)\log(e))$ operations in $\mathbb{K}$.*
- *Conversely, Chinese remaindering modulo polynomials with sum of degrees d can be done in $\mathcal{O}(\mathsf{M}(d)\log(d))$ operations in $\mathbb{K}$.*

Proof. For the first and third point, we refer the reader to (von zur Gathen and Gerhard, 2013, Chapters 9 and 10). For the second point, we first compute $q = q_1 \cdots q_s$ in time $\mathcal{O}(\mathsf{M}(e)\log(e))$, reduce p modulo q in time $\mathcal{O}(\mathsf{M}(d + e))$, and use the simultaneous modular reduction algorithm of von zur Gathen and Gerhard (2013, Corollary 10.17), which takes time $\mathcal{O}(\mathsf{M}(e)\log(e))$. Besides, we have $\mathsf{M}(d + e) + \mathsf{M}(e)\log(e) \in \mathcal{O}(\mathsf{M}(d) + \mathsf{M}(e)\log(e))$, as can be seen by considering the cases $d \leqslant e$ and $d > e$. □

6.2. Main algorithm

For a Jordan matrix $\mathbf{J} \in \mathbb{K}^{\sigma\times\sigma}$ given in standard representation, and for any x in $\mathbb{K}$, we will denote by $\mathrm{rep}(x, \mathbf{J})$ the number of pairs (x, s) appearing in that representation, counting repetitions (so that $\sum_{x\in\mathbb{K}} \mathrm{rep}(x, \mathbf{J}) = \sigma$).

For an integer $k \in \{0, \ldots, \lceil\log(\sigma)\rceil\}$, we select from the representation of $\mathbf{J}$ all those pairs (x, s) with s in $\{2^k, \ldots, 2^{k+1} - 1\}$, obtaining a set $\mathbf{J}^{(k)}$. Since $\mathbf{J}$ is in standard representation, we can compute all $\mathbf{J}^{(k)}$ by a single pass through the array $\mathbf{J}$, and we can ensure for free that all $\mathbf{J}^{(k)}$ themselves are in standard representation. We decompose $\mathbf{J}^{(k)}$ further into two classes $\mathbf{J}^{(k,>m)}$, where all pairs (x, s) are such that $\mathrm{rep}(x, \mathbf{J}^{(k)})$ is greater than m, and $\mathbf{J}^{(k,\leqslant m)}$, which contains all other pairs. As above, this decomposition can be done in linear time, and we can ensure for no extra cost that $\mathbf{J}^{(k,>m)}$ and $\mathbf{J}^{(k,\leqslant m)}$ are in standard representation. Explicitly, these sequences will be written as

$$\mathbf{J}^{(k,>m)} = ((x_1^{(k)}, s_{1,1}^{(k)}), \ldots, (x_1^{(k)}, s_{1,r_1^{(k)}}^{(k)}), \ldots, (x_{t^{(k)}}^{(k)}, s_{t^{(k)},1}^{(k)}), \ldots, (x_{t^{(k)}}^{(k)}, s_{t^{(k)},r_{t^{(k)}}^{(k)}}^{(k)})),$$

with $(r_i^{(k)})_i = (\mathrm{rep}(x_i^{(k)}, \mathbf{J}^{(k)}))_i$ non-increasing, and where for i in $\{1, \ldots, t^{(k)}\}$, $r_i^{(k)} > m$ and $(s_{i,j}^{(k)})_j$ is a non-increasing sequence of elements in $\{2^k, \ldots, 2^{k+1} - 1\}$. The corresponding sets of columns in the input matrix $\mathbf{E}$ and the output $\mathbf{F}$ will be written

$$\mathbf{E}^{(k,>m)} = (\mathbf{E}_{i,j}^{(k,>m)})_{1 \leqslant i \leqslant t^{(k)}, 1 \leqslant j \leqslant r_i^{(k)}}$$

and

$$\mathbf{F}^{(k,>m)} = (\mathbf{F}_{i,j}^{(k,>m)})_{1 \leqslant i \leqslant t^{(k)}, 1 \leqslant j \leqslant r_i^{(k)}};$$

they will be treated using a direct application of Lemma 6.2. Similarly, we write

$$\mathbf{J}^{(k,\leqslant m)} = ((\xi_1^{(k)}, \sigma_{1,1}^{(k)}), \ldots, (\xi_1^{(k)}, \sigma_{1,\rho_1^{(k)}}^{(k)}), \ldots, (\xi_{\tau^{(k)}}^{(k)}, \sigma_{\tau^{(k)},1}^{(k)}), \ldots, (\xi_{\tau^{(k)}}^{(k)}, \sigma_{\tau^{(k)},\rho_{\tau^{(k)}}^{(k)}}^{(k)})),$$

with $(\rho_i^{(k)})_i = (\mathrm{rep}(\xi_i^{(k)}, \mathbf{J}^{(k)}))_i$ non-increasing, and where for i in $\{1, \ldots, \tau^{(k)}\}$, $\rho_i^{(k)} \leqslant m$ and $(\sigma_{i,j}^{(k)})_j$ is a non-increasing sequence of elements in $\{2^k, \ldots, 2^{k+1} - 1\}$. The corresponding sets of columns in the input matrix $\mathbf{E}$ and the output $\mathbf{F}$ will be written $\mathbf{E}^{(k,\leqslant m)}$ and $\mathbf{F}^{(k,\leqslant m)}$; more precisely, they take the form

$$\mathbf{E}^{(k,\leqslant m)} = (\mathbf{E}_{i,j}^{(k,\leqslant m)})_{1 \leqslant i \leqslant \tau^{(k)}, 1 \leqslant j \leqslant \rho_i^{(k)}}$$

and

$$\mathbf{F}^{(k,\leqslant m)} = (\mathbf{F}_{i,j}^{(k,\leqslant m)})_{1 \leqslant i \leqslant \tau^{(k)}, 1 \leqslant j \leqslant \rho_i^{(k)}},$$

and will be treated using a Chinese remaindering approach.

In the main loop, the index k will range from 0 to $\lfloor \log(\sigma/m) \rfloor$. After that stage, all entries (x, s) in $\mathbf{J}$ that were not processed yet are such that $s > \sigma/m$. In particular, if we call $\mathbf{J}^{(\infty,\leqslant m)}$ the set of these remaining entries, we deduce that this set has cardinality at most m; thus $\mathrm{rep}(x, \mathbf{J}^{(\infty,\leqslant m)}) \leqslant m$ holds for all x and we process these entries using the Chinese remaindering approach.

Algorithm ComputeResiduals constructs all these sets $\mathbf{J}^{(k,>m)}$, $\mathbf{J}^{(k,\leqslant m)}$, and $\mathbf{J}^{(\infty,\leqslant m)}$, then extracts the corresponding columns from $\mathbf{E}$ (this is the subroutine ExtractColumns), and processes these subsets of columns, before merging all the results.

Algorithm 5 *(*ComputeResiduals*).*

Input:

- a Jordan matrix $\mathbf{J}$ in $\mathbb{K}^{\sigma \times \sigma}$ in standard representation,
- a matrix $\mathbf{P} \in \mathbb{K}[X]^{m \times m}$,
- a matrix $\mathbf{E} \in \mathbb{K}^{m \times \sigma}$.

Output: the product $\mathbf{P} \cdot \mathbf{E} \in \mathbb{K}^{m \times \sigma}$.

1. For k from 0 to $\lfloor \log(\sigma/m) \rfloor$
  a. $\mathbf{J}^{(k)} \leftarrow ((x, s) \in \mathbf{J} \mid 2^k \leqslant s < 2^{k+1})$
  b. $\mathbf{J}^{(k,>m)} \leftarrow ((x, s) \in \mathbf{J}^{(k)} \mid \mathrm{rep}(x, \mathbf{J}^{(k)}) > m)$
  c. $\mathbf{E}^{(k,>m)} \leftarrow$ ExtractColumns($\mathbf{E}, \mathbf{J}^{(k,>m)}$)
  d. $\mathbf{F}^{(k,>m)} \leftarrow$ ComputeResidualsByShiftingP($\mathbf{J}^{(k,>m)}, \mathbf{P}, \mathbf{E}^{(k,>m)}$)
  e. $\mathbf{J}^{(k,\leqslant m)} \leftarrow ((x, s) \in \mathbf{J}^{(k)} \mid \mathrm{rep}(x, \mathbf{J}^{(k)}) \leqslant m)$
  f. $\mathbf{E}^{(k,\leqslant m)} \leftarrow$ ExtractColumns($\mathbf{E}, \mathbf{J}^{(k,\leqslant m)}$)
  g. $\mathbf{F}^{(k,\leqslant m)} \leftarrow$ ComputeResidualsByCRT($\mathbf{J}^{(k,\leqslant m)}, \mathbf{P}, \mathbf{E}^{(k,\leqslant m)}$)
2. $\mathbf{J}^{(\infty,\leqslant m)} \leftarrow ((x, s) \in \mathbf{J} \mid 2^{\lfloor \log(\sigma/m) \rfloor + 1} \leqslant s)$
3. $\mathbf{E}^{(\infty,\leqslant m)} \leftarrow$ ExtractColumns($\mathbf{E}, \mathbf{J}^{(\infty,\leqslant m)}$)
4. $\mathbf{F}^{(\infty,\leqslant m)} \leftarrow$ ComputeResidualsByCRT($\mathbf{J}^{(\infty,\leqslant m)}, \mathbf{P}, \mathbf{E}^{(\infty,\leqslant m)}$)
5. Return Merge($(\mathbf{F}^{(k,>m)})_{0 \leqslant k \leqslant \lfloor \log(\sigma/m) \rfloor}, (\mathbf{F}^{(k,\leqslant m)})_{0 \leqslant k \leqslant \lfloor \log(\sigma/m) \rfloor}, \mathbf{F}^{(\infty,\leqslant m)}$)

6.2.1. Computing the residual by shifting **P**

We start with the case of the sets $\mathbf{J}^{(k,>m)}$, for which we follow a direct approach. Below, recall that we write

$$\mathbf{J}^{(k,>m)} = ((x_1^{(k)}, s_{1,1}^{(k)}), \ldots, (x_1^{(k)}, s_{1,r_1^{(k)}}^{(k)}), \ldots, (x_{t^{(k)}}^{(k)}, s_{t^{(k)},1}^{(k)}), \ldots, (x_{t^{(k)}}^{(k)}, s_{t^{(k)},r_{t^{(k)}}^{(k)}}^{(k)})),$$

with $s_{i,1}^{(k)} \geqslant s_{i,j}^{(k)}$ for any k, i, and j. For a fixed k, we compute $\mathbf{P}_i^{(k)} = \mathbf{P}(X + x_i^{(k)}) \bmod X^{s_{i,1}^{(k)}}$, for i in $\{1, \ldots, t^{(k)}\}$, and do the corresponding matrix products. This is described in Algorithm 6; we give below a bound on the total time spent in this algorithm, that is, for all k in $\{0, \ldots, \lfloor \log(\sigma/m) \rfloor\}$. Before that, we give two lemmas: the first one will allow us to control the cost of the calculations in this case; in the second one, we explain how to efficiently compute the polynomial matrices $\mathbf{P}_i^{(k)}$.

Lemma 6.4. *The following bound holds:*

$$\sum_{k=0}^{\lfloor \log(\sigma/m) \rfloor} \sum_{i=1}^{t^{(k)}} r_i^{(k)} s_{i,1}^{(k)} \in \mathcal{O}(\sigma).$$

Proof. By construction, we have the estimate

$$\sum_{k=0}^{\lfloor \log(\sigma/m) \rfloor} \sum_{i=1}^{t^{(k)}} \sum_{j=1}^{r_i^{(k)}} s_{i,j}^{(k)} \leqslant \sigma,$$

since this represents the total size of all blocks contained in the sequences $\mathbf{J}^{(k,>m)}$. Now, for fixed k and i, the construction of $\mathbf{J}^{(k)}$ implies that the inequality $s_{i,1}^{(k)} \leqslant 2 s_{i,j}^{(k)}$ holds for all j. This shows that we have

$$r_i^{(k)} s_{i,1}^{(k)} \leqslant 2 \sum_{j=1}^{r_i^{(k)}} s_{i,j}^{(k)},$$

and the conclusion follows by summing over all k and i. □

In the following lemma, we explain how to compute the polynomial matrices $\mathbf{P}_i^{(k)}$ in an efficient manner, for i in $\{1, \ldots, t^{(k)}\}$ and for all the values of k we need.

Lemma 6.5. *Suppose that the sum of the row degrees of* **P** *is* $\mathcal{O}(\sigma)$*. Then one can compute the matrices* $\mathbf{P}_i^{(k)}$ *for all* k *in* $\{0, \ldots, \lfloor \log(\sigma/m) \rfloor\}$ *and* i *in* $\{1, \ldots, t^{(k)}\}$ *using* $\mathcal{O}(m \, \mathsf{M}(\sigma) \log(\sigma))$ *operations in* $\mathbb{K}$*.*

Proof. We use the second item in Lemma 6.3 to first compute $\mathbf{P} \bmod (X - x_i^{(k)})^{s_{i,1}^{(k)}}$, for all k and i as in the statement of the lemma. Here, the sum of the degrees is

$$S = \sum_{k,i} s_{i,1}^{(k)},$$

so we get a total cost of $\mathcal{O}(\mathsf{M}(d) + \mathsf{M}(S) \log(S))$ for an entry of **P** of degree d. Summing over all entries, and using the fact that the sum of the row degrees of **P** is $\mathcal{O}(\sigma)$, we obtain a total cost of

$$\mathcal{O}(m \, \mathsf{M}(\sigma) + m^2 \mathsf{M}(S) \log(S)).$$

Now, because we consider here $\mathbf{J}^{(k,>m)}$, we have $r_i^{(k)} > m$ for all k and i. Hence, using the super-linearity of $\mathsf{M}(\cdot)$, the term $m^2 \mathsf{M}(S) \log(S)$ admits the upper bound

Algorithm 6 *(ComputingResidualsByShiftingP).*

Input:

- $\mathbf{J}^{(k,>m)} = ((x_1^{(k)}, s_{1,1}^{(k)}), \dots, (x_1^{(k)}, s_{1,r_1^{(k)}}^{(k)}), \dots, (x_{t^{(k)}}^{(k)}, s_{t^{(k)},1}^{(k)}), \dots, (x_{t^{(k)}}^{(k)}, s_{t^{(k)},r_{t^{(k)}}^{(k)}}^{(k)}))$ in standard representation,
- a matrix $\mathbf{P} \in \mathbb{K}[X]^{m\times m}$,
- a matrix $\mathbf{E}^{(k,>m)} = [\mathbf{E}_{1,1}^{(k,>m)} | \cdots | \mathbf{E}_{t^{(k)},r_{t^{(k)}}^{(k)}}^{(k,>m)}] \in \mathbb{K}^{m\times \sum_{i,j} s_{i,j}^{(k)}}$ with $\mathbf{E}_{i,j}^{(k,>m)} \in \mathbb{K}^{m\times s_{i,j}^{(k)}}$ for all i, j.

Output: the product $\mathbf{P}\cdot\mathbf{E}^{(k,>m)} \in \mathbb{K}^{m\times \sum_{i,j} s_{i,j}^{(k)}}$.

1. $(\mathbf{P}_i^{(k)})_{1\leqslant i\leqslant t^{(k)}} \leftarrow (\mathbf{P}(X+x_i) \bmod X^{s_{i,1}^{(k)}})_{1\leqslant i\leqslant t^{(k)}}$

2. For i from 1 to $t^{(k)}$

  a. $(\mathbf{E}_{i,j,\mathrm{poly}}^{(k,>m)})_{1\leqslant j\leqslant r_i^{(k)}} \leftarrow (\mathbf{E}_{i,j}^{(k,>m)}[1, X, \dots, X^{s_{i,j}^{(k)}-1}]^\mathsf{T})_{1\leqslant j\leqslant r_i^{(k)}}$

  b. $[\mathbf{F}_{i,1,\mathrm{poly}}^{(k,>m)} | \cdots | \mathbf{F}_{i,r_i^{(k)},\mathrm{poly}}^{(k,>m)}] \leftarrow \mathbf{P}_i^{(k)} [\mathbf{E}_{i,1,\mathrm{poly}}^{(k,>m)} | \cdots | \mathbf{E}_{i,r_i^{(k)},\mathrm{poly}}^{(k,>m)}]$

  c. For j from 1 to $r_i^{(k)}$, $\mathbf{F}_{i,j}^{(k,>m)} \leftarrow (\mathrm{coeff}(\mathbf{F}_{i,j,\mathrm{poly}}^{(k,>m)}, \ell))_{0\leqslant \ell < s_{i,j}^{(k)}}$

3. Return $[\mathbf{F}_{1,1}^{(k,>m)} | \cdots | \mathbf{F}_{t^{(k)},r_{t^{(k)}}^{(k)}}^{(k,>m)}]$

$$m\,\mathsf{M}\left(\sum_{k,i} r_i^{(k)} s_{i,1}^{(k)}\right)\log(S),$$

which is in $\mathcal{O}(m\,\mathsf{M}(\sigma)\log(\sigma))$ in view of Lemma 6.4.

Then we apply a variable shift to all these polynomials to replace X by $X + x_i^{(k)}$. Using the first item in Lemma 6.3, for fixed k and i, the cost is $\mathcal{O}(m^2\mathsf{M}(s_{i,1}^{(k)})\log(s_{i,1}^{(k)}))$. Hence, the total time is again $\mathcal{O}(m^2\mathsf{M}(S)\log(S))$, so the same overall bound as above holds. □

Lemma 6.6. *Algorithm 6 is correct. Given the polynomial matrices computed in Lemma 6.5, the total time spent in this algorithm for all k in* $\{0, \dots, \lfloor\log(\sigma/m)\rfloor\}$ *is* $\mathcal{O}(\mathsf{MM}(m, \sigma/m))$ *operations in* $\mathbb{K}$.

Proof. Correctness of the algorithm follows from Lemma 6.2, so we focus on the cost analysis.

Lemma 6.5 gives the cost of computing all polynomial matrices needed at Step **1**. The only other arithmetic operations are those done in the matrix products at Step **2.b**: we multiply matrices of respective sizes $m \times m$ and $m \times r_i^{(k)}$, with entries of degree less than $s_{i,1}^{(k)}$. For given k and i, since we have $m < r_i^{(k)}$, the cost is $\mathcal{O}(\mathsf{MM}(m, s_{i,1}^{(k)}) r_i^{(k)}/m)$; using the super-linearity of $d \mapsto \mathsf{MM}(m, d)$, this is in $\mathcal{O}(\mathsf{MM}(m, r_i^{(k)} s_{i,1}^{(k)}/m))$. Applying again Lemma 6.4, we deduce that the sum over all k and i is $\mathcal{O}(\mathsf{MM}(m, \sigma/m))$. □

6.2.2. Computing the residual by Chinese remaindering

The second case to consider is $\mathbf{J}^{(k,\leqslant m)}$. Recall that for a given index k, we write this sequence as

$$\mathbf{J}^{(k,\leqslant m)} = ((\xi_1^{(k)}, \sigma_{1,1}^{(k)}), \dots, (\xi_1^{(k)}, \sigma_{1,\rho_1^{(k)}}^{(k)}), \dots, (\xi_{\tau^{(k)}}^{(k)}, \sigma_{\tau^{(k)},1}^{(k)}), \dots, (\xi_{\tau^{(k)}}^{(k)}, \sigma_{\tau^{(k)},\rho_{\tau^{(k)}}^{(k)}}^{(k)})),$$

with $\rho_{\tau^{(k)}}^{(k)} \leqslant \cdots \leqslant \rho_1^{(k)} \leqslant m$ for all i in $\{1, \dots, \tau^{(k)}\}$. In this case, $\tau^{(k)}$ may be large so the previous approach may lead us to compute too many matrices $\mathbf{P}_i^{(k)}$. Instead, for fixed k and j, we use Chinese remaindering to transform the corresponding submatrices $\mathbf{E}_{i,j}^{(k,\leqslant m)}$ into a polynomial matrix $\mathbf{E}_j^{(k,\leqslant m)}$ of small column dimension; this allows us to efficiently perform matrix multiplication by $\mathbf{P}$ on the left, and we eventually get $\mathbf{P}\cdot\mathbf{E}_{i,j}^{(k,\leqslant m)}$ by computing the first coefficients in a Taylor expansion of this product around every $\xi_i^{(k)}$.

To simplify the notation in the algorithm, we also suppose that for a fixed k, the points $\xi_1^{(k)}, \ldots, \xi_{\tau^{(k)}}^{(k)}$ all appear the same number of times in $\mathbf{J}^{(k,\leqslant m)}$. This is done by replacing $\rho_1^{(k)}, \ldots, \rho_{\tau^{(k)}}^{(k)}$ by their maximum $\rho_1^{(k)}$ (simply written $\rho^{(k)}$ in the pseudo-code) and adding suitable blocks $(\xi_i^{(k)}, \sigma_{i,j}^{(k)})$, with all new $\sigma_{i,j}^{(k)}$ set to zero.

Algorithm 7 *(ComputingResidualsByCRT).*

Input:

- $\mathbf{J}^{(k,\leqslant m)} = ((\xi_1^{(k)}, \sigma_{1,1}^{(k)}), \ldots, (\xi_1^{(k)}, \sigma_{1,\rho^{(k)}}^{(k)}), \ldots, (\xi_{\tau^{(k)}}^{(k)}, \sigma_{\tau^{(k)},1}^{(k)}), \ldots, (\xi_{\tau^{(k)}}^{(k)}, \sigma_{\tau^{(k)},\rho^{(k)}}^{(k)}))$ in standard representation,
- a matrix $\mathbf{P} \in \mathbb{K}[X]^{m\times m}$,
- a matrix $\mathbf{E}^{(k,\leqslant m)} = [\mathbf{E}_{1,1}^{(k,\leqslant m)} | \cdots | \mathbf{E}_{\tau^{(k)},\rho^{(k)}}^{(k,\leqslant m)}] \in \mathbb{K}^{m\times \sum_{i,j}\sigma_{i,j}^{(k)}}$ with $\mathbf{E}_{i,j}^{(k,\leqslant m)} \in \mathbb{K}^{m\times \sigma_{i,j}^{(k)}}$ for all i, j.

Output: the product $\mathbf{P}\cdot\mathbf{E}^{(k,\leqslant m)} \in \mathbb{K}^{m\times\sum_{i,j}\sigma_{i,j}^{(k)}}$.

1. For j from 1 to $\rho^{(k)}$
   - **a.** $(\mathbf{E}_{i,j,\text{shifted}}^{(k,\leqslant m)})_{1\leqslant i\leqslant \tau^{(k)}} \leftarrow (\mathbf{E}_{i,j}^{(k,\leqslant m)}[1, X-\xi_i^{(k)}, \ldots, (X-\xi_i^{(k)})^{\sigma_{i,j}^{(k)}-1}]^{\mathsf{T}})_{1\leqslant i\leqslant\tau^{(k)}}$
   - **b.** $\mathbf{E}_{j,\text{shifted}}^{(k,\leqslant m)} \leftarrow \mathrm{CRT}((\mathbf{E}_{i,j,\text{shifted}}^{(k,\leqslant m)})_{1\leqslant i\leqslant\tau^{(k)}}, ((X-\xi_i^{(k)})^{\sigma_{i,j}^{(k)}})_{1\leqslant i\leqslant\tau^{(k)}})$
2. $[\mathbf{F}_{1,\text{shifted}}^{(k,\leqslant m)} | \cdots | \mathbf{F}_{\rho^{(k)},\text{shifted}}^{(k,\leqslant m)}] \leftarrow \mathbf{P}[\mathbf{E}_{1,\text{shifted}}^{(k,\leqslant m)} | \cdots | \mathbf{E}_{\rho^{(k)},\text{shifted}}^{(k,\leqslant m)}]$
3. For j from 1 to $\rho^{(k)}$
   - **a.** $(\mathbf{F}_{i,j,\text{shifted}}^{(k,\leqslant m)})_{1\leqslant i\leqslant\tau^{(k)}} \leftarrow (\mathbf{F}_{j,\text{shifted}}^{(k,\leqslant m)} \bmod (X-\xi_i^{(k)})^{\sigma_{i,j}^{(k)}})_{1\leqslant i\leqslant\tau^{(k)}}$
   - **b.** $\mathbf{F}_{i,j}^{(k,\leqslant m)} \leftarrow (\mathrm{coeff}(\mathbf{F}_{i,j,\text{shifted}}^{(k,\leqslant m)}(X+\xi_i^{(k,\leqslant m)}), \ell))_{0\leqslant \ell<\sigma_{i,j}^{(k)}}$
4. Return $[\mathbf{F}_{1,1}^{(k,\leqslant m)} | \cdots | \mathbf{F}_{\tau^{(k)},\rho^{(k)}}^{(k,\leqslant m)}]$

Lemma 6.7. *Algorithm 7 is correct. If the sum of the row degrees of* $\mathbf{P}$ *is in* $\mathcal{O}(\sigma)$*, the total time spent in this algorithm for all k in* $\{0, \ldots, \lfloor\log(\sigma/m)\rfloor, \infty\}$ *is*

$$\mathcal{O}(\mathsf{MM}(m, \sigma/m)\log(\sigma/m) + m\mathsf{M}(\sigma)\log(\sigma))$$

operations in $\mathbb{K}$.

Proof. Proving correctness amounts to verifying that we compute the quantities described in Lemma 6.2. Indeed, the formulas in the algorithm show that for all k, i, j, we have $\mathbf{F}_{i,j,\text{shifted}}^{(k,\leqslant m)} = \mathbf{P}\mathbf{E}_{i,j,\text{shifted}}^{(k,\leqslant m)} \bmod (X-\xi_i^{(k)})^{\sigma_{i,j}^{(k)}}$; the link with Lemma 6.2 is made by observing that $\mathbf{E}_{i,j,\text{shifted}}^{(k,\leqslant m)} = \mathbf{E}_{i,j,\text{poly}}^{(k,\leqslant m)}(X-\xi_i^{(k)})$ and $\mathbf{F}_{i,j,\text{shifted}}^{(k,\leqslant m)} = \mathbf{F}_{i,j,\text{poly}}^{(k,\leqslant m)}(X-\xi_i^{(k)})$.

In terms of complexity, the first item in Lemma 6.3 shows that for a given index k, Step **1.a** can be done in time

$$\mathcal{O}\left(m\sum_{i,j}\mathsf{M}\left(\sigma_{i,j}^{(k)}\right)\log\left(\sigma_{i,j}^{(k)}\right)\right),$$

for a total cost of $\mathcal{O}(m\mathsf{M}(\sigma)\log(\sigma))$. Step **1.b** can be done in quasi-linear time as well: for each k and j, we can compute each of the m entries of the polynomial vector $\mathbf{E}_{j,\text{shifted}}^{(k,\leqslant m)}$ by fast Chinese remaindering (third item in Lemma 6.3), using

$$\mathcal{O}\left(\mathsf{M}\left(S_j^{(k)}\right)\log\left(S_j^{(k)}\right)\right)$$

operations in $\mathbb{K}$, with $S_j^{(k)} = \sum_i \sigma_{i,j}^{(k)}$. Taking all rows into account, and summing over all indices k and j, we obtain again a total cost of $\mathcal{O}(m\mathsf{M}(\sigma)\log(\sigma))$.

The next step to examine is the polynomial matrix product at Step **2**. The matrix $\mathbf{P}$ has size $m \times m$, and the sum of its row degrees is by assumption $\mathcal{O}(\sigma)$; using the partial linearization technique presented in Section 4, we can replace $\mathbf{P}$ by a matrix of size $\mathcal{O}(m) \times m$ with entries of degree at most σ/m.

For a fixed choice of k, the right-hand side has size $m \times \rho^{(k)}$, and its columns have respective degrees less than $S_1^{(k)}, \ldots, S_{\rho^{(k)}}^{(k)}$. We split each of its columns into new columns of degree at most σ/m, so that the jth column is split into $\mathcal{O}(1 + S_j^{(k)} m/\sigma)$ columns (the constant term 1 dominates when $S_j^{(k)} \leqslant \sigma/m$). Thus, the new right-hand side has $\mathcal{O}(\rho^{(k)} + (S_1^{(k)} + \cdots + S_{\rho^{(k)}}^{(k)})m/\sigma)$ columns and degree at most σ/m.

Now, taking all k into account, we remark that the left-hand side remains the same; thus, we are led to do one matrix product with degrees σ/m, with left-hand side of size $\mathcal{O}(m) \times m$, and right-hand side having column dimension at most

$$\sum_{k \in \{0,\ldots,\lfloor \log(\sigma/m) \rfloor\} \cup \{\infty\}} \rho^{(k)} + \frac{(S_1^{(k)} + \cdots + S_{\rho^{(k)}}^{(k)})m}{\sigma}.$$

Since all $\rho^{(k)}$ are at most m, the first term sums up to $\mathcal{O}(m \log(\sigma/m))$; by construction, the second one adds up to $\mathcal{O}(m)$. Hence, the matrix product we need can be done in time $\mathcal{O}(\mathsf{MM}(m, \sigma/m) \log(\sigma/m))$.

For a given k, $\mathbf{F}_{1,\text{shifted}}^{(k,\leqslant m)}, \ldots, \mathbf{F}_{\rho^{(k)},\text{shifted}}^{(k,\leqslant m)}$ are vectors of size m. Furthermore, for each j the entries of $\mathbf{F}_{j,\text{shifted}}^{(k,\leqslant m)}$ have degree less than $S_1^{(k)} + d_1, \ldots, S_m^{(k)} + d_m$ respectively, where $d_1, \ldots, d_m$ are the degrees of the rows of $\mathbf{P}$. In particular, for a fixed k, the reductions at Step **3.a** can be done in time

$$\mathcal{O}\left(\rho^{(k)}(\mathsf{M}(d_1 + \cdots + d_m)) + m \sum_{j=1}^{\rho^{(k)}} \mathsf{M}(S_j^{(k)}) \log(S_j^{(k)})\right)$$

using fast multiple reduction, by means of the second item in Lemma 6.3. Using our assumption on $\mathbf{P}$, and the fact that $\rho^{(k)} \leqslant m$, we see that the first term is $\mathcal{O}(m\mathsf{M}(\sigma))$, which adds up to $\mathcal{O}(m\mathsf{M}(\sigma)\log(\sigma/m))$ if we sum over k. The second term adds up to $\mathcal{O}(m\mathsf{M}(\sigma)\log(\sigma))$, as was the case for Step **1.b**.

The same analysis is used for the shifts taking place at Step **3.b** as for those in Step **1.a**: for fixed k and j, the cost is $\mathcal{O}(m\mathsf{M}(S_j^{(k)})\log(S_j^{(k)}))$, and we conclude as above. □

7. Minimal interpolation basis via linearization

In this section, we give an efficient algorithm based on linearization techniques to compute interpolation bases for the case of an arbitrary matrix $\mathbf{J}$ and an arbitrary shift; in particular, we prove Theorem 1.4.

In addition to being interesting on its own, the algorithm in this section allows us to handle the base cases in the recursion of the divide-and-conquer algorithm presented in Section 3. For that particular case, we have $m/2 \leqslant \sigma \leqslant m$; the algorithm we give here solves this base case using $\mathcal{O}(m^\omega \log(m))$ operations in $\mathbb{K}$.

Proposition 7.1. *Let* $\mathbf{J} \in \mathbb{K}^{\sigma \times \sigma}$, $\mathbf{E} \in \mathbb{K}^{m \times \sigma}$, $\mathbf{s} \in \mathbb{N}^m$, *and let* $\delta \in \mathbb{N}$ *be a bound on the degree of the minimal polynomial of* $\mathbf{J}$. *Then, Algorithm 9 solves Problem 1 deterministically, using*

$\mathcal{O}(\sigma^\omega(\lceil m/\sigma \rceil + \log(\delta)))$ *if* $\omega > 2$

$\mathcal{O}(\sigma^2(\lceil m/\sigma \rceil + \log(\delta))\log(\sigma))$ *if* $\omega = 2$

operations in $\mathbb{K}$; it returns the unique interpolation basis $\mathbf{P}$ *for* $(\mathbf{E}, \mathbf{J})$ *which is in* $\mathbf{s}$*-Popov form. Besides, the maximal degree in* $\mathbf{P}$ *is at most* δ *and the sum of the* column *degrees of* $\mathbf{P}$ *is at most* σ.

We remark that the degree of the minimal polynomial of $\mathbf{J}$ is at most σ. In Algorithm 9, we require that δ be a power of 2 and thus we may have $\delta > \sigma$; still we can always choose $\delta < 2\sigma$. The proof is deferred until Subsection 7.4, where we also recall the definition of the shifted Popov form (Beckermann et al., 2006).

To obtain this result, we rely on linear algebra tools via the use of the linearization in Beckermann and Labahn (2000), where an interpolant is seen as a linear relation between the rows of a striped Krylov matrix. The reader may also refer to (Kailath, 1980, §6.3 and §6.4) for a presentation of this point of view. In Beckermann and Labahn (2000), it is assumed that $\mathbf{J}$ is upper triangular: this yields recurrence relations (Beckermann and Labahn, 2000, Theorem 6.1), leading to an iterative algorithm (Beckermann and Labahn, 2000, Algorithm FFFG) to compute an interpolation basis in shifted Popov form in a fraction-free way.

Here, to obtain efficiency and deal with a general $\mathbf{J}$, we proceed in two steps. First, we compute the row rank profile of the striped Krylov matrix $\mathcal{K}$ with an algorithm *à la* (Keller-Gehrig, 1985), which uses at most $\log(\delta)$ steps and supports different orderings of the rows in $\mathcal{K}$ depending on the input shift. Then, we use the resulting independent rows of $\mathcal{K}$ to compute the specific rows in the nullspace of $\mathcal{K}$ which correspond to the interpolation basis in shifted Popov form.

We note that when $\sigma = \mathcal{O}(1)$, the cost bound in Proposition 7.1 is linear in m, while the dense representation of the output $m \times m$ polynomial matrix will use at least m^2 field elements. We will see in Subsection 7.4 that when $\sigma < m$, at least $m - \sigma$ columns of the basis in $\mathbf{s}$-Popov form have only one nonzero coefficient which is 1, and thus those columns can be described without involving any arithmetic operation. Hence, the actual computation is restricted to an $m \times \sigma$ submatrix of the output basis.

7.1. Linearization

Our goal is to explain how to transform the problem of finding interpolants into a problem of linear algebra over $\mathbb{K}$. This will involve a straightforward linearization of the polynomials in the output interpolation basis $\mathbf{P}$, expanding them as a list of coefficients so that $\mathbf{P}$ is represented as a matrix over $\mathbb{K}$. Correspondingly, we show how from the input $(\mathbf{E}, \mathbf{J})$ one can build a matrix $\mathcal{K}$ over $\mathbb{K}$ which is such that an interpolant for $(\mathbf{E}, \mathbf{J})$ corresponds to a vector in the left nullspace of $\mathcal{K}$. Then, since we will be looking for interpolants that have a small degree with respect to the column shifts given by $\mathbf{s}$, we describe a way to adapt these constructions so that they facilitate taking into account the influence of $\mathbf{s}$. This gives a first intuition of some properties of the linearization of an interpolant that has small shifted degree: this will then be presented in details in Subsection 7.2.

Let us first describe the linearization of interpolants, which are seen as row vectors in $\mathbb{K}[X]^{1\times m}$. In what follows, we suppose that we know a bound $\delta \in \mathbb{N}_{>0}$ on the degree of the minimal polynomial of $\mathbf{J}$; one can always choose $\delta = \sigma$. In Subsection 7.2, we will exhibit $\mathbf{s}$-minimal interpolation bases for $(\mathbf{E}, \mathbf{J})$ whose entries all have degree at most δ (while in general such a basis may have degree up to $\delta + |\mathbf{s} - \min(\mathbf{s})|$). Thus, in this Section 7, we focus on solutions to Problem 1 that have degree at most δ. Correspondingly, $\mathbb{K}[X]_{\leqslant\delta}$ denotes the set of polynomials in $\mathbb{K}[X]$ of degree at most δ.

Given $\mathbf{P} \in \mathbb{K}[X]_{\leqslant\delta}^{n\times m}$ for some $n \geqslant 1$, we write it as a polynomial of matrices: $\mathbf{P} = \mathbf{P}_0 + \mathbf{P}_1 X + \cdots + \mathbf{P}_\delta X^\delta$ where each $\mathbf{P}_j$ is a scalar matrix in $\mathbb{K}^{n\times m}$; then the *expansion* of $\mathbf{P}$ (in degree δ) is the matrix $\mathcal{E}(\mathbf{P}) = [\mathbf{P}_0 \mid \mathbf{P}_1 \mid \cdots \mid \mathbf{P}_\delta] \in \mathbb{K}^{n\times m(\delta+1)}$. The reciprocal operation is called *compression* (in degree δ): given a scalar matrix $\mathbf{M} \in \mathbb{K}^{n\times m(\delta+1)}$, we write it with blocks $\mathbf{M} = [\mathbf{M}_0 \mid \mathbf{M}_1 \mid \cdots \mid \mathbf{M}_\delta]$ where each $\mathbf{M}_j$ is in $\mathbb{K}^{n\times m}$, and then we define its compression as $\mathcal{C}(\mathbf{M}) = \mathbf{M}_0 + \mathbf{M}_1 X + \cdots + \mathbf{M}_\delta X^\delta \in \mathbb{K}[X]_{\leqslant\delta}^{n\times m}$. These definitions of $\mathcal{E}(\mathbf{P})$ and $\mathcal{C}(\mathbf{M})$ hold for any row dimension n; this n will always be clear from the context.

Now, given some matrices $\mathbf{E} \in \mathbb{K}^{m\times\sigma}$ and $\mathbf{J} \in \mathbb{K}^{\sigma\times\sigma}$, our interpolation problem asks to find $\mathbf{P} \in \mathbb{K}[X]_{\leqslant\delta}^{m\times m}$ such that $\mathbf{P} \cdot \mathbf{E} = 0$. Writing $\mathbf{P} = \mathbf{P}_0 + \mathbf{P}_1 X + \cdots + \mathbf{P}_\delta X^\delta$, we recall that $\mathbf{P} \cdot \mathbf{E} = \mathbf{P}_0\mathbf{E} + \mathbf{P}_1\mathbf{E}\mathbf{J} + \cdots + \mathbf{P}_\delta\mathbf{E}\mathbf{J}^\delta$. Then, in accordance to the linearization of $\mathbf{P}$, the input $(\mathbf{E}, \mathbf{J})$ is expanded as follows:

$$\mathcal{K}(\mathbf{E}) = \begin{bmatrix} \mathbf{E} \\ \hline \mathbf{EJ} \\ \hline \vdots \\ \hline \mathbf{EJ}^{\delta} \end{bmatrix} \in \mathbb{K}^{m(\delta+1)\times\sigma}.$$

This way, we have $\mathbf{P} \cdot \mathbf{E} = \mathcal{E}(\mathbf{P})\mathcal{K}(\mathbf{E})$ for any polynomial matrix $\mathbf{P} \in \mathbb{K}[X]^{n\times m}_{\leqslant\delta}$. In particular, a row vector $\mathbf{p} \in \mathbb{K}[X]^{1\times m}_{\leqslant\delta}$ is an interpolant for $\mathbf{E}$ if and only if $\mathcal{E}(\mathbf{p})\mathcal{K}(\mathbf{E}) = 0$, that is, $\mathcal{E}(\mathbf{p})$ is in the (left) nullspace of $\mathcal{K}(\mathbf{E})$. Up to some permutation of the rows and different degree constraints, this so-called *striped-Krylov* matrix $\mathcal{K}(\mathbf{E})$ was used in Beckermann and Labahn (2000) for the purpose of computing interpolants.

Notation. For the rest of this Section 7, we will use the letter i for rows of $\mathcal{K}(\mathbf{E})$ and for columns of $\mathcal{E}(\mathbf{P})$; the letter j for columns of $\mathcal{K}(\mathbf{E})$; the letter d for the block of columns of $\mathcal{E}(\mathbf{P})$ which correspond to coefficients of degree d in $\mathbf{P}$, as well as for the corresponding block $\mathbf{EJ}^d$ of rows of $\mathcal{K}(\mathbf{E})$; the letter c for the columns of this degree d block in $\mathcal{E}(\mathbf{P})$ and for the rows of the block $\mathbf{EJ}^d$ in $\mathcal{K}(\mathbf{E})$.

Example 7.2 *(Linearization).* In this example, we have $m = \sigma = \delta = 3$ and the base field is the finite field with 97 elements; the input matrices are

$$\mathbf{E} = \begin{bmatrix} 27 & 49 & 29 \\ 50 & 58 & 0 \\ 77 & 10 & 29 \end{bmatrix} \quad \text{and} \quad \mathbf{Z} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Then, we have

$$\mathcal{K}(\mathbf{E}) = \begin{bmatrix} 27 & 49 & 29 \\ 50 & 58 & 0 \\ 77 & 10 & 29 \\ 0 & 27 & 49 \\ 0 & 50 & 58 \\ 0 & 77 & 10 \\ 0 & 0 & 27 \\ 0 & 0 & 50 \\ 0 & 0 & 77 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

It is easily checked that $\mathbf{p}_1 = (-1, -1, 1) \in \mathbb{K}[X]^m$ is an interpolant for $(\mathbf{E}, \mathbf{Z})$, since $\mathbf{E}_{3,*} = \mathbf{E}_{1,*} + \mathbf{E}_{2,*}$. Other interpolants are for example $\mathbf{p}_2 = (3X + 13,\ X + 57,\ 0)$ which has row degree 1, $\mathbf{p}_3 = \left(X^2 + 36X,\ 31X,\ 0\right)$ which has row degree 2, and $\mathbf{p}_4 = \left(X^3,\ 0,\ 0\right)$ which has row degree 3. We have

$$\begin{aligned}
\mathcal{E}(\mathbf{p}_1) &= [\ 96\ 96\ 1 \mid 0\ \ 0\ \ 0 \mid 0\ 0\ 0 \mid 0\ 0\ 0\ ] \\
\mathcal{E}(\mathbf{p}_2) &= [\ 13\ 57\ 0 \mid 3\ \ 1\ \ 0 \mid 0\ 0\ 0 \mid 0\ 0\ 0\ ] \\
\mathcal{E}(\mathbf{p}_3) &= [\ 0\ \ 0\ \ 0 \mid 36\ 31\ 0 \mid 1\ 0\ 0 \mid 0\ 0\ 0\ ] \\
\mathcal{E}(\mathbf{p}_4) &= [\ 0\ \ 0\ \ 0 \mid 0\ \ 0\ \ 0 \mid 0\ 0\ 0 \mid 1\ 0\ 0\ ].
\end{aligned}$$

Besides, one can check that the matrix

$$\mathbf{P} = \begin{bmatrix} X^2 + 36X & 31X & 0 \\ 3X + 13 & X + 57 & 0 \\ 96 & 96 & 1 \end{bmatrix},$$

whose rows are $(\mathbf{p}_3, \mathbf{p}_2, \mathbf{p}_1)$ is a reduced basis for the module $\mathfrak{I}(\mathbf{E}, \mathbf{Z})$ of Hermite–Padé approximants of order 3 for $\mathbf{f} = \left(29X^2 + 49X + 27,\ 58X + 50,\ 29X^2 + 10X + 77\right)$. $\square$

Now, we need tools to interpret the $\mathbf{s}$-minimality of an interpolation basis. In Example 7.2, we see that $\mathbf{p}_1$ has $\mathbf{0}$-row degree 0 and therefore appears in $\mathbf{P}$; however $\mathbf{p}_4$ has $\mathbf{0}$-row degree 3 and does not appear in $\mathbf{P}$. On the other hand, considering $\mathbf{s} = (0, 3, 6)$, the $\mathbf{s}$-row degree of $\mathbf{p}_4$ is 3, while the one of $\mathbf{p}_1$ is 6: when forming rows of a $(0, 3, 6)$-minimal interpolation basis, $\mathbf{p}_4$ is a better candidate than $\mathbf{p}_1$. We see through this example that the uniform shift $\mathbf{s} = \mathbf{0}$ leads to look in priority for relations involving the first rows of the matrix $\mathcal{K}(\mathbf{E})$; on the other hand, the shift $\mathbf{s} = (0, 3, 6)$ leads to look for relations involving in priority the rows $\mathbf{E}_{1,*}$, $\mathbf{E}_{1,*}\mathbf{Z}$, $\mathbf{E}_{1,*}\mathbf{Z}^2$, and $\mathbf{E}_{1,*}\mathbf{Z}^3$ in $\mathcal{K}(\mathbf{E})$ before considering the rows $\mathbf{E}_{2,*}$ and $\mathbf{E}_{3,*}$.

Going back to the general case, we define a notion of *priority* of the row c of $\mathbf{E}\mathbf{J}^d$ in $\mathcal{K}(\mathbf{E})$. Let $\mathbf{v} \in \mathbb{K}^{1\times m(\delta+1)}$ be any relation between the rows of $\mathcal{K}(\mathbf{E})$ involving this row, meaning that, writing $\mathbf{p} = \mathbf{p}_0 + \mathbf{p}_1 X + \cdots + \mathbf{p}_\delta X^\delta = \mathcal{C}(\mathbf{v})$ for the corresponding interpolant, the coefficient in column c of $\mathbf{p}_d$ is nonzero. This implies that the $\mathbf{s}$-row degree of $\mathbf{p}$ is at least $s_c + d$. Since the $\mathbf{s}$-row degree is precisely what we want to minimize in order to obtain an $\mathbf{s}$-minimal interpolation basis, the priority of the rows of $\mathcal{K}(\mathbf{E})$ can be measured by the function $\psi_{\mathbf{s}}$ defined by $\psi_{\mathbf{s}}(c, d) = s_c + d$. Then, when computing relations between rows of $\mathcal{K}(\mathbf{E})$, we should use in priority the rows with low $\psi_{\mathbf{s}}(d, r)$ in order to get interpolants with small $\mathbf{s}$-row degree.

To take this into account, we extend the linearization framework by using a permutation of the rows of $\mathcal{K}(\mathbf{E})$ so that they appear in non-increasing order of their priority given by $\mathbf{s}$. This way, an interpolant with *small* $\mathbf{s}$-row degree is always one whose expansion forms a relation between the *first* rows of the permuted $\mathcal{K}(\mathbf{E})$. To preserve properties such as $\mathbf{P} \cdot \mathbf{E} = \mathcal{E}(\mathbf{P})\mathcal{K}(\mathbf{E})$, we naturally permute the columns of $\mathcal{E}(\mathbf{P})$ accordingly. If $\ell = [\psi_{\mathbf{s}}(1, 0), \ldots, \psi_{\mathbf{s}}(m, 0), \psi_{\mathbf{s}}(1, 1), \ldots, \psi_{\mathbf{s}}(m, 1), \ldots, \psi_{\mathbf{s}}(1, \delta), \ldots, \psi_{\mathbf{s}}(m, \delta)]$ in $\mathbb{Z}^{1\times m(\delta+1)}$ denotes the row vector indicating the priorities of the rows of $\mathcal{K}(\mathbf{E})$, then we choose an $m(\delta + 1) \times m(\delta + 1)$ permutation matrix $\pi_{\mathbf{s}}$ such that the list $\ell\pi_{\mathbf{s}}$ is non-decreasing. Then, the matrix $\pi_{\mathbf{s}}^{-1}\mathcal{K}(\mathbf{E})$ is the matrix $\mathcal{K}(\mathbf{E})$ with rows permuted so that they are arranged by non-increasing priority, that is, by non-decreasing values of $\psi_{\mathbf{s}}$. Furthermore, the permutation $\pi_{\mathbf{s}}$ induces a bijection $\phi_{\mathbf{s}}$ which keeps track of the position changes when applying the permutation: it associates to (c, d) the index $\phi_{\mathbf{s}}(c, d)$ of the element $\psi_{\mathbf{s}}(c, d)$ in the sorted list $\ell\pi_{\mathbf{s}}$. We now give precise definitions.

Definition 7.3 *(Priority).* Let $\mathbf{E} \in \mathbb{K}^{m\times\sigma}$ and $\mathbf{J} \in \mathbb{K}^{\sigma\times\sigma}$, and let $\mathbf{s} \in \mathbb{N}^m$. The priority function $\psi_{\mathbf{s}} : \{1, \ldots, m\} \times \{0, \ldots, \delta\} \to \mathbb{Z}$ is defined by $\psi_{\mathbf{s}}(c, d) = s_c + d$. Let $\ell = [\psi_{\mathbf{s}}(1, 0), \ldots, \psi_{\mathbf{s}}(m, 0), \psi_{\mathbf{s}}(1, 1), \ldots, \psi_{\mathbf{s}}(m, 1), \ldots, \psi_{\mathbf{s}}(1, \delta), \ldots, \psi_{\mathbf{s}}(m, \delta)]$ be the sequence of priorities in $\mathbb{Z}^{1\times m(\delta+1)}$. Then, we define $\pi_{\mathbf{s}}$ as the unique permutation matrix in $\mathbb{K}^{m(\delta+1)\times m(\delta+1)}$ along with the corresponding indexing function

$$\phi_{\mathbf{s}} : \{1, \ldots, m\} \times \{0, \ldots, \delta\} \to \{1, \ldots, m(\delta + 1)\}$$
$$[\phi_{\mathbf{s}}(1, 0), \ldots, \phi_{\mathbf{s}}(m, 0), \ldots, \phi_{\mathbf{s}}(1, \delta), \ldots, \phi_{\mathbf{s}}(m, \delta)] = [1, 2, \ldots, m(\delta + 1)]\, \pi_{\mathbf{s}}$$

which are such that

(i) $\ell\pi_{\mathbf{s}}$ is non-decreasing;
(ii) whenever $(c, d) \neq (c', d')$ are such that $\psi_{\mathbf{s}}(c, d) = \psi_{\mathbf{s}}(c', d')$, we have $c \neq c'$ and assuming without loss of generality that $c < c'$, then $\phi_{\mathbf{s}}(c, d) < \phi_{\mathbf{s}}(c', d')$.

Besides, we define $\mathcal{K}_{\mathbf{s}}(\mathbf{E}) = \pi_{\mathbf{s}}^{-1}\mathcal{K}(\mathbf{E})$ as well as the shifted expansion $\mathcal{E}_{\mathbf{s}}(\mathbf{P}) = \mathcal{E}(\mathbf{P})\pi_{\mathbf{s}}$ and the shifted compression $\mathcal{C}_{\mathbf{s}}(\mathbf{M}) = \mathcal{C}(\mathbf{M}\pi_{\mathbf{s}})$.

In other words, $\pi_{\mathbf{s}}$ is the unique permutation which lexicographically sorts the sequence

$$[(\psi_{\mathbf{s}}(1, 0), 1), \ldots, (\psi_{\mathbf{s}}(m, 0), m), \ldots, (\psi_{\mathbf{s}}(1, \delta), 1), \ldots, (\psi_{\mathbf{s}}(m, \delta), m)].$$

A representation of $\pi_{\mathbf{s}}$ can be computed using $\mathcal{O}(m\delta \log(m\delta))$ integer comparisons, and a representation of $\phi_{\mathbf{s}}$ can be computed using its definition in time linear in $m(\delta + 1)$. In the specific case of the uniform shift $\mathbf{s} = \mathbf{0}$, we have $\psi_{\mathbf{s}}(c, d) = d$, $\pi_{\mathbf{s}}$ is the identity matrix, and $\phi_{\mathbf{s}}(c, d) = c + md$, and we have the identities $\mathcal{K}_{\mathbf{s}}(\mathbf{E}) = \mathcal{K}(\mathbf{E})$, $\mathcal{C}_{\mathbf{s}}(\mathbf{M}) = \mathcal{C}(\mathbf{M})$, $\mathcal{E}_{\mathbf{s}}(\mathbf{P}) = \mathcal{E}(\mathbf{P})$. The main ideas of the rest of Section 7 can be understood focusing on this particular case.

Example 7.4 *(Input linearization, continued).* In the context of Example 7.2, if we consider the shifts $\mathbf{s} = (0, 3, 6)$ and $\mathbf{t} = (3, 0, 2)$, then we have

$$\mathcal{K}_{\mathbf{s}}(\mathbf{E}) = \begin{bmatrix} 27 & 49 & 29 \\ 0 & 27 & 49 \\ 0 & 0 & 27 \\ 0 & 0 & 0 \\ 50 & 58 & 0 \\ 0 & 50 & 58 \\ 0 & 0 & 50 \\ 0 & 0 & 0 \\ 77 & 10 & 29 \\ 0 & 77 & 10 \\ 0 & 0 & 77 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathcal{K}_{\mathbf{t}}(\mathbf{E}) = \begin{bmatrix} 50 & 58 & 0 \\ 0 & 50 & 58 \\ 0 & 0 & 50 \\ 77 & 10 & 29 \\ 27 & 49 & 29 \\ 0 & 0 & 0 \\ 0 & 77 & 10 \\ 0 & 27 & 49 \\ 0 & 0 & 77 \\ 0 & 0 & 27 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Besides, one can check that the shifted expansions of the interpolants $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, and $\mathbf{p}_4$ with respect to $\mathbf{s}$ and $\mathbf{t}$ are

$$\begin{aligned}
\mathcal{E}_{\mathbf{s}}(\mathbf{p}_1) &= [\; 96 \;\; 0 \;\; 0 \;\; 0 \;\; 96 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \;] \\
\mathcal{E}_{\mathbf{t}}(\mathbf{p}_1) &= [\; 96 \;\; 0 \;\; 0 \;\; 1 \;\; 96 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;] \\
\mathcal{E}_{\mathbf{s}}(\mathbf{p}_2) &= [\; 13 \;\; 3 \;\; 0 \;\; 0 \;\; 57 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;] \\
\mathcal{E}_{\mathbf{t}}(\mathbf{p}_2) &= [\; 57 \;\; 1 \;\; 0 \;\; 0 \;\; 13 \;\; 0 \;\; 0 \;\; 3 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;] \\
\mathcal{E}_{\mathbf{s}}(\mathbf{p}_3) &= [\; 0 \;\; 36 \;\; 1 \;\; 0 \;\; 0 \;\; 31 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;] \\
\mathcal{E}_{\mathbf{t}}(\mathbf{p}_3) &= [\; 0 \;\; 31 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 36 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;] \\
\mathcal{E}_{\mathbf{s}}(\mathbf{p}_4) &= [\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;] \\
\mathcal{E}_{\mathbf{t}}(\mathbf{p}_4) &= [\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;]. \quad \square
\end{aligned}$$

7.2. Minimal linear relations and minimal interpolation bases

From the previous subsection, the intuition is that the minimality of interpolants can be read on the corresponding linear relations between the rows of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$, as the fact that they involve in priority the first rows. Here, we support this intuition with rigorous statements, presenting a notion of minimality for linear relations between the rows of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$, and showing that an $\mathbf{s}$-minimal interpolation basis for $(\mathbf{E}, \mathbf{J})$ corresponds to a specific set of m such minimal relations.

First we show that, given a polynomial row vector and a degree shift, one can directly read the pivot index (Mulders and Storjohann, 2003, Section 2) of the vector from its expansion. Extending the definitions in Mulders and Storjohann (2003) to the shifted case, we define the $\mathbf{s}$-pivot index, $\mathbf{s}$-pivot entry, and $\mathbf{s}$-pivot degree of a nonzero row vector as follows.

Definition 7.5 *(Pivot).* Let $\mathbf{p} = [p_c]_c \in \mathbb{K}[X]^{1\times m}$ be a nonzero row vector and let $\mathbf{s} \in \mathbb{N}^m$ be a degree shift. The $\mathbf{s}$-*pivot index* of $\mathbf{p}$ is the largest column index $c \in \{1, \ldots, m\}$ such that $\deg(p_c) + s_c$ is equal to the $\mathbf{s}$-row degree $\mathrm{rdeg}_{\mathbf{s}}(\mathbf{p})$ of this row; then, p_c and $\deg(p_c)$ are called the $\mathbf{s}$-*pivot entry* and the $\mathbf{s}$-*pivot degree* of $\mathbf{p}$, respectively.

The following result will be useful for our purpose, since pivot indices can be used to easily identify some specific forms of reduced polynomial matrices.

Lemma 7.6. *Let $\mathbf{p} = [p_c]_c \in \mathbb{K}[X]^{1\times m}_{\leqslant \delta}$ be a nonzero row vector. Then, $i = \phi_{\mathbf{s}}(c, d)$ is the column index of the rightmost nonzero coefficient in $\mathcal{E}_{\mathbf{s}}(\mathbf{p})$ if and only if $\mathbf{p}$ has $\mathbf{s}$-pivot index c and $\mathbf{s}$-pivot degree $\deg(p_c) = d$.*

Proof. We distinguish three sets of entries of $\mathcal{E}_{\mathbf{s}}(\mathbf{p})$ with column index $\geqslant i$: the one at index i, the ones that have a higher $\psi_{\mathbf{s}}$, and the ones that have the same $\psi_{\mathbf{s}}$:

- if the coefficient at index $i = \phi_{\mathbf{s}}(c,d)$ in $\mathcal{E}_{\mathbf{s}}(\mathbf{p})$ is nonzero then $\deg(p_c) \geqslant d$, and if $\deg(p_c) = d$ then the coefficient at index $i = \phi_{\mathbf{s}}(c,d)$ in $\mathcal{E}_{\mathbf{s}}(\mathbf{p})$ is nonzero;
- the coefficient at index $\phi_{\mathbf{s}}(c',d')$ in $\mathcal{E}_{\mathbf{s}}(\mathbf{p})$ is zero for all (c',d') such that $\psi_{\mathbf{s}}(c',d') > \psi_{\mathbf{s}}(c,d)$ if and only if $s_{c'} + \deg(p_{c'}) \leqslant s_c + d$ for all $1 \leqslant c' \leqslant m$;
- assuming $s_{c'} + \deg(p_{c'}) \leqslant s_c + d$ for all $1 \leqslant c' \leqslant m$, the coefficient at index $\phi_{\mathbf{s}}(c',d')$ in $\mathcal{E}_{\mathbf{s}}(\mathbf{p})$ is zero for all (c',d') such that $\psi_{\mathbf{s}}(c',d') = \psi_{\mathbf{s}}(c,d)$ with $\phi_{\mathbf{s}}(c',d') > i = \phi_{\mathbf{s}}(c,d)$ (by definition of $\pi_{\mathbf{s}}$, this implies $c' > c$) if and only if we have $s_{c'} + \deg(p_{c'}) < s_c + d$ for all $c' > c$;

these three points prove the equivalence. □

We have seen that an interpolant for $(\mathbf{E},\mathbf{J})$ corresponds to a linear relation between the rows of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$. From this perspective, the preceding result implies that an interpolant with $\mathbf{s}$-pivot index c and $\mathbf{s}$-pivot degree d corresponds to a linear relation which expresses the row at index $\phi_{\mathbf{s}}(c,d)$ in $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ as a linear combination of the rows at indices smaller than $\phi_{\mathbf{s}}(c,d)$. Now, we give a precise correspondence between minimal interpolation bases and sets of linear relations which involve in priority the first rows of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$.

Example 7.7 *(Minimal relations).* Let us consider the context of Example 7.2 with the uniform shift. As mentioned above, the matrix $\mathbf{P}$ whose rows are $(\mathbf{p}_3, \mathbf{p}_2, \mathbf{p}_1)$ is a minimal interpolation basis. The pivot indices of $\mathbf{p}_3, \mathbf{p}_2, \mathbf{p}_1$ are 1, 2, 3, and their pivot degrees are 2, 1, 0. Besides, we remark that

- the relation $\mathcal{E}(\mathbf{p}_1)$ involves the row $c = 3$ of $\mathbf{E}$ and the rows above this one in $\mathcal{K}(\mathbf{E})$;
- $\mathcal{E}(\mathbf{p}_2)$ involves the row $c = 2$ of $\mathbf{EZ}$ and the rows above this one in $\mathcal{K}(\mathbf{E})$, and there is no linear relation involving the row $c = 2$ of $\mathbf{E}$ and the rows above it in $\mathcal{K}(\mathbf{E})$;
- $\mathcal{E}(\mathbf{p}_3)$ involves the row $c = 1$ of $\mathbf{EZ}^2$ and the rows above it in $\mathcal{K}(\mathbf{E})$: one can check that there is no linear relation between the row $c = 1$ of $\mathbf{EZ}$ and the rows above it in $\mathcal{K}(\mathbf{E})$. □

This example suggests that we can give a link between the minimal row degree of a minimal interpolation basis and some minimal exponent δ_c such that the row c of the block $\mathbf{EJ}^{\delta_c}$ is a linear combination on the rows above it in $\mathcal{K}(\mathbf{E})$. Extending this to the case of any shift $\mathbf{s}$ leads us to the following definition, which is reminiscent of the so-called *minimal indices* (or *Kronecker indices*) for $\mathbf{0}$-minimal nullspace bases (Kailath, 1980, Section 6.5.4).

Definition 7.8 *(Minimal degree).* Let $\mathbf{J} \in \mathbb{K}^{\sigma\times\sigma}$ and $\mathbf{E} \in \mathbb{K}^{m\times\sigma}$, and let $\mathbf{s} \in \mathbb{N}^m$. The $\mathbf{s}$-*minimal degree* of $(\mathbf{E},\mathbf{J})$ is the tuple $(\delta_1,\ldots,\delta_m)$ where for each $c \in \{1,\ldots,m\}$, $\delta_c \in \mathbb{N}$ is the smallest exponent such that the row ${\mathbf{EJ}^{\delta_c}}_{c,*} = \mathcal{K}_{\mathbf{s}}(\mathbf{E})_{\phi_{\mathbf{s}}(c,\delta_c),*}$ is a linear combination of the rows in $\{\mathcal{K}_{\mathbf{s}}(\mathbf{E})_{i,*}, i < \phi_{\mathbf{s}}(c,\delta_c)\}$.

We note that we have $\delta_c \leqslant \delta$ for every c, since the minimal polynomial of the matrix $\mathbf{J} \in \mathbb{K}^{\sigma\times\sigma}$ is of degree at most δ. We now state in Lemma 7.9 and Corollary 7.11 that the minimal degree of $(\mathbf{E},\mathbf{J})$ indeed corresponds to a notion of minimality of interpolants and interpolation bases. Until the end of this Subsection 7.2, we fix a matrix $\mathbf{E} \in \mathbb{K}^{m\times\sigma}$ and a matrix $\mathbf{J} \in \mathbb{K}^{\sigma\times\sigma}$.

Lemma 7.9. *Let* $\mathbf{p} = [p_1,\ldots,p_m] \in \mathbb{K}[X]^{1\times m}_{\leqslant\delta}$, $\mathbf{s} \in \mathbb{N}^m$, *and* c *in* $\{1,\ldots,m\}$. *If* $\mathbf{p}$ *is an interpolant for* $(\mathbf{E},\mathbf{J})$ *with* $\mathbf{s}$*-pivot index* c, *then* $\mathbf{p}$ *has* $\mathbf{s}$*-pivot degree* $\deg(p_c) \geqslant \delta_c$. *Besides, there is an interpolant* $\mathbf{p} \in \mathbb{K}[X]^{1\times m}_{\leqslant\delta}$ *for* $(\mathbf{E},\mathbf{J})$ *which has* $\mathbf{s}$*-pivot index* c *and* $\mathbf{s}$*-pivot degree* $\deg(p_c) = \delta_c$.

Proof. First, assume $\mathbf{p}$ is an interpolant with $\mathbf{s}$-pivot index c, and let $d = \deg(p_c)$ be the degree of the $\mathbf{s}$-pivot entry of $\mathbf{p}$. According to Lemma 7.6, the rightmost nonzero element of $\mathcal{E}_{\mathbf{s}}(\mathbf{p})$ is at index $\phi_{\mathbf{s}}(c,d)$, and since $\mathbf{p}$ is an interpolant for $(\mathbf{E},\mathbf{J})$ we have $\mathcal{E}_{\mathbf{s}}(\mathbf{P})\mathcal{K}_{\mathbf{s}}(\mathbf{E}) = 0$. This implies that the row $\phi_{\mathbf{s}}(c,d)$ of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ is a linear combination of the rows in $\{\mathcal{K}_{\mathbf{s}}(\mathbf{E})_{i,*}, i < \phi_{\mathbf{s}}(c,d)\}$, which in turn implies $d \geqslant \delta_c$ by definition of δ_c. Now, the definition of δ_c also ensures that the row $\phi_{\mathbf{s}}(c,\delta_c)$ of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ is a linear combination of the rows $\{\mathcal{K}_{\mathbf{s}}(\mathbf{E})_{i,*}, i < \phi_{\mathbf{s}}(c,d_c)\}$. This linear combination forms a vector $\mathbf{v}$ in the nullspace of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ with its rightmost nonzero element at index $\phi_{\mathbf{s}}(c, \delta_c)$; then by Lemma 7.6, $\mathbf{p} = \mathcal{C}_{\mathbf{s}}(\mathbf{v})$ is an interpolant with $\mathbf{s}$-pivot index c and $\mathbf{s}$-pivot degree δ_c. Besides, $\mathbf{p}$ has degree at most δ by construction. □

Now, we want to extend these considerations on row vectors and interpolants to matrices and interpolation bases. In connection with the notion of pivot of a row, there is a specific form of reduced matrices called the weak Popov form (Mulders and Storjohann, 2003), for which we extend the definition to any shift $\mathbf{s}$ as follows.

Definition 7.10 *(weak Popov form, pivot degree).* Let $\mathbf{P}$ in $\mathbb{K}[X]^{m\times m}$ have full rank, and let $\mathbf{s}$ in $\mathbb{N}^m$. Then, $\mathbf{P}$ is said to be in $\mathbf{s}$-*weak Popov form* if the $\mathbf{s}$-pivot indices of its rows are pairwise distinct. Furthermore, the $\mathbf{s}$-*pivot degree* of $\mathbf{P}$ is the tuple $(d_1, \ldots, d_m) \in \mathbb{N}^m$ where for $c \in \{1, \ldots, m\}$, d_c is the $\mathbf{s}$-pivot degree of the row of $\mathbf{P}$ which has $\mathbf{s}$-pivot index c.

A matrix $\mathbf{P}$ in $\mathbf{s}$-weak Popov form is in particular $\mathbf{s}$-reduced. Then, Lemma 7.9 leads to the following result; we remark that even though the matrix $\mathbf{P}$ in this corollary is $\mathbf{s}$-reduced and each of its rows is an interpolant, we do not yet claim that it is an interpolation basis.

Corollary 7.11. *There is a matrix $\mathbf{P} \in \mathbb{K}[X]^{m\times m}_{\leqslant \delta}$ in $\mathbf{s}$-weak Popov form, with $\mathbf{s}$-pivot entries on the diagonal and $\mathbf{s}$-pivot degree $(\delta_1, \ldots, \delta_m)$, such that every row of $\mathbf{P}$ is an interpolant for $(\mathbf{E}, \mathbf{J})$.*

Proof. For every c in $\{1, \ldots, m\}$, Lemma 7.9 shows that there is an interpolant $\mathbf{p}_c$ for $(\mathbf{E}, \mathbf{J})$ which has degree at most δ, has $\mathbf{s}$-pivot index c, and has $\mathbf{s}$-pivot degree δ_c. Then, considering the matrix $\mathbf{P}$ in $\mathbb{K}[X]^{m\times m}$ whose row c is $\mathbf{p}_c$ gives the conclusion. □

We conclude this section by proving that the $\mathbf{s}$-minimal degree of $(\mathbf{E}, \mathbf{J})$ is directly linked to the $\mathbf{s}$-row degree of a minimal interpolation basis, which proves in particular that the matrix $\mathbf{P}$ in Corollary 7.11 is an $\mathbf{s}$-minimal interpolation basis for $(\mathbf{E}, \mathbf{J})$.

Lemma 7.12. *Let $\mathbf{A} \in \mathbb{K}[X]^{m\times m}$ be $\mathbf{s}$-reduced such that each row of $\mathbf{A}$ is an interpolant for $(\mathbf{E}, \mathbf{J})$. Then, $\mathbf{A}$ is an interpolation basis for $(\mathbf{E}, \mathbf{J})$ if and only if the $\mathbf{s}$-row degree $\mathrm{rdeg}_{\mathbf{s}}(\mathbf{A})$ of $\mathbf{A}$ is $(s_1 + \delta_1, \ldots, s_m + \delta_m)$ up to a permutation of the rows of $\mathbf{A}$. In particular, if $\mathbf{P}$ is a matrix as in Corollary 7.11, then $\mathbf{P}$ is an $\mathbf{s}$-minimal interpolation basis for $(\mathbf{E}, \mathbf{J})$.*

Proof. We denote $\mathbf{P} \in \mathbb{K}[X]^{m\times m}_{\leqslant \delta}$ a matrix as in Corollary 7.11; $\mathbf{P}$ is in particular $\mathbf{s}$-reduced and has $\mathbf{s}$-row degree exactly $(s_1 + \delta_1, \ldots, s_m + \delta_m)$.

First, we assume that $\mathbf{A}$ is an interpolation basis for $(\mathbf{E}, \mathbf{J})$. Remarking that a matrix $\mathbf{B}$ is in $\mathbf{s}$-weak Popov form if and only if $\mathbf{B}\mathbf{X}^{\mathbf{s}}$ is in weak Popov form, we know from Mulders and Storjohann (2003, Section 2) that $\mathbf{A}$ is left-unimodularly equivalent to a matrix $\mathbf{B}$ in $\mathbf{s}$-weak Popov form. Besides, up to a permutation of the rows of $\mathbf{B}$, we assume without loss of generality that the pivot entries $\mathbf{B}$ are on the diagonal. Then, denoting its $\mathbf{s}$-pivot degree by $(d_1, \ldots, d_m)$, the $\mathbf{s}$-row degree of $\mathbf{B}$ is $(s_1 + d_1, \ldots, s_m + d_m)$. Since $\mathbf{A}$ and $\mathbf{B}$ are $\mathbf{s}$-reduced and unimodularly equivalent, they have the same $\mathbf{s}$-row degree up to a permutation (Kailath, 1980, Lemma 6.3-14): thus it is enough to prove that $(d_1, \ldots, d_m) = (\delta_1, \ldots, \delta_m)$. By Lemma 7.9 applied to each row of $\mathbf{B}$, $d_1, \ldots, d_m$ are at least $\delta_1, \ldots, \delta_m$, respectively. On the other hand, since $\mathbf{B}$ is an interpolation basis, there is a nonsingular matrix $\mathbf{U} \in \mathbb{K}[X]^{m\times m}$ such that $\mathbf{P} = \mathbf{U}\mathbf{B}$. Since $\mathbf{P}$ is $\mathbf{s}$-reduced with the $\mathbf{s}$-row degree $(s_1 + \delta_1, \ldots, s_m + \delta_m)$, we have $\deg(\det(\mathbf{P})) = |\mathrm{rdeg}_{\mathbf{s}}(\mathbf{P})| - |\mathbf{s}| = \delta_1 + \cdots + \delta_m$ (Zhou, 2012, Lemma 2.10). Similarly, we have $\deg(\det(\mathbf{B})) = |\mathrm{rdeg}_{\mathbf{s}}(\mathbf{B})| - |\mathbf{s}| = d_1 + \cdots + d_m$. Considering the determinantal degree in the identity $\mathbf{P} = \mathbf{U}\mathbf{B}$ yields $\delta_1 + \cdots + \delta_m \geqslant d_1 + \cdots + d_m$, from which we conclude $d_c = \delta_c$ for all $1 \leqslant c \leqslant m$.

Now, we note that this also implies that the determinant of $\mathbf{U}$ is constant, thus $\mathbf{U}$ is unimodular and consequently $\mathbf{P}$ is an interpolation basis: since $\mathbf{P}$ is $\mathbf{s}$-reduced by construction, it is an $\mathbf{s}$-minimal interpolation basis.

Finally, we assume that $\mathbf{A}$ has $\mathbf{s}$-row degree $(s_1+\delta_1,\dots,s_m+\delta_m)$ up to a permutation. Since $\mathbf{P}$ is an interpolation basis, there is a nonsingular matrix $\mathbf{U}\in\mathbb{K}[X]^{m\times m}$ such that $\mathbf{A}=\mathbf{UP}$. Since $\mathbf{A}$ is $\mathbf{s}$-reduced, we have $\deg(\det(\mathbf{A}))=|\mathrm{rdeg}_{\mathbf{s}}(\mathbf{A})|-|\mathbf{s}|=\delta_1+\cdots+\delta_m$. Then, considering the determinantal degree in the identity $\mathbf{A}=\mathbf{UP}$ shows that the determinant of $\mathbf{U}$ is a nonzero constant, that is, $\mathbf{U}$ is unimodular. Thus, $\mathbf{A}$ is an interpolation basis. □

Remark 7.13. As can be observed in the definition of the $\mathbf{s}$-minimal degree and in the proof of Lemma 7.9, one can use Gaussian elimination on the rows of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ to build each row of the $\mathbf{s}$-minimal interpolation basis $\mathbf{P}$. This gives a method for solving Problem 1 using linear algebra. Then, the main goal in the rest of this section is to show how to perform the computation of $\mathbf{P}$ efficiently. □

7.3. Row rank profile and minimal degree

The reader may have noted that $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ has $m\sigma(\delta+1)$ coefficients in $\mathbb{K}$, and in general $m\sigma\delta$ may be beyond our target cost bound given in Proposition 7.1. Here, we show that one can focus on a small subset of the rows of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ which contains enough information to compute linear relations leading to a matrix $\mathbf{P}$ as in Corollary 7.11. Then, we present a fast algorithm to compute this subset of rows. We also use these results to bound the average $\mathbf{s}$-row degree of any $\mathbf{s}$-minimal interpolation basis.

To begin with, we give some helpful structure properties of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$, which will be central in choosing the subset of rows and in the designing a fast algorithm which computes independent rows in $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ without having to consider the whole matrix.

Lemma 7.14 *(Structure of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$). Let $\mathbf{J}\in\mathbb{K}^{\sigma\times\sigma}$ and $\mathbf{E}\in\mathbb{K}^{m\times\sigma}$, and let $\mathbf{s}\in\mathbb{N}^m$. Let $\phi_{\mathbf{s}},\psi_{\mathbf{s}}$ be as in Definition 7.3.*

- *For each c, $d\mapsto\phi_{\mathbf{s}}(c,d)$ is strictly increasing.*
- *If (c,d) and (c',d') are such that $\phi_{\mathbf{s}}(c,d)<\phi_{\mathbf{s}}(c',d')$, then for any $k\leqslant\min(\delta-d,\delta-d')$ we have $\phi_{\mathbf{s}}(c,d+k)<\phi_{\mathbf{s}}(c',d'+k)$.*
- *Suppose that for some $i\in\{1,\dots,m(\delta+1)\}$, the row at index i in $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ is a linear combination of the rows of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ with indices in $\{1,\dots,i-1\}$. Then, writing $i=\phi_{\mathbf{s}}(c,d)$, for every $d'\in\{0,\dots,\delta-d\}$ the row at index $i'=\phi_{\mathbf{s}}(c,d+d')$ in $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ is a linear combination of the rows of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ with indices in $\{1,\dots,i'-1\}$.*

Proof. The first item is clear since $\psi_{\mathbf{s}}(c,d)=s_c+d$. For the second item, we consider two cases. First, if $\psi_{\mathbf{s}}(c,d)<\psi_{\mathbf{s}}(c',d')$, this means $s_c+d<s_{c'}+d'$ from which we obviously obtain $\psi_{\mathbf{s}}(c,d+k)<\psi_{\mathbf{s}}(c',d'+k)$, and in particular $\phi_{\mathbf{s}}(c,d+k)<\phi_{\mathbf{s}}(c',d'+k)$. Second, if $\psi_{\mathbf{s}}(c,d)=\psi_{\mathbf{s}}(c',d')$, we must have $c<c'$ by choice of $\phi_{\mathbf{s}}$, and then we also have $\psi_{\mathbf{s}}(c,d+k)=\psi_{\mathbf{s}}(c',d'+k)$ with $c<c'$ which implies that $\phi_{\mathbf{s}}(c,d+k)<\phi_{\mathbf{s}}(c',d'+k)$.

The third item is a direct rewriting in the linearization framework of the following property. Let $\mathbf{p}\in\mathbb{K}[X]^{1\times m}_{\leqslant\delta}$ be an interpolant for $(\mathbf{E},\mathbf{J})$ with $\mathbf{s}$-pivot index c and $\mathbf{s}$-row degree d, and consider $d'\leqslant\delta-d$; then the row vector $X^{d'}\mathbf{p}$, with entries of degree more than δ taken modulo the minimal polynomial of $\mathbf{J}$, is an interpolant for $(\mathbf{E},\mathbf{J})$ with $\mathbf{s}$-pivot index c and $\mathbf{s}$-row degree $d+d'$. □

We remark that, when choosing a subset of $r=\mathrm{rank}(\mathcal{K}_{\mathbf{s}}(\mathbf{E}))$ linearly independent rows in $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$, all other rows in the matrix are linear combinations of those. Because our goal is to find relations which involve in priority the first rows of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$, we are specifically interested in the *first* r independent rows in $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$. More precisely, we focus on the *row rank profile* $(i_1,\dots,i_r)$ of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$, that is, the lexicographically smallest tuple with entries in $\{1,\dots,m(\delta+1)\}$ such that the submatrix of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ formed by the rows with indices in $\{i_1,\dots,i_r\}$ has rank $r=\mathrm{rank}(\mathcal{K}_{\mathbf{s}}(\mathbf{E}))$. Then, for each c the row $\mathbf{E}\mathbf{J}^{\delta_c}{}_{c,*}=\mathcal{K}_{\mathbf{s}}(\mathbf{E})_{\phi_{\mathbf{s}}(c,\delta_c),*}$ is a linear combination of the rows in $\{\mathcal{K}_{\mathbf{s}}(\mathbf{E})_{i_k,*},1\leqslant k\leqslant r\}$. We now show that this row rank profile is directly related to the $\mathbf{s}$-minimal degree of $(\mathbf{E},\mathbf{J})$.

Lemma 7.15. *Let* $\mathbf{J} \in \mathbb{K}^{\sigma\times\sigma}$ *and* $\mathbf{E} \in \mathbb{K}^{m\times\sigma}$, *let* $\mathbf{s} \in \mathbb{N}^m$, *and let* $(i_1, \ldots, i_r)$ *be the row rank profile of* $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$. *For* $k \in \{1, \ldots, r\}$, *we write* $i_k = \phi_{\mathbf{s}}(c_k, d_k)$ *for some (unique)* $1 \leqslant c_k \leqslant m$ *and* $0 \leqslant d_k \leqslant \delta$. *Given* $c \in \{1, \ldots, m\}$,

- *for all* $0 \leqslant d < \delta_c$ *we have* $\phi_{\mathbf{s}}(c, d) \in \{i_1, \ldots, i_r\}$,
- $\delta_c = 0$ *if and only if* $c_k \neq c$ *for all* $k \in \{1, \ldots, r\}$,
- *if* $\delta_c > 0$ *we have* $\delta_c = 1 + \max\{d_k \mid 1 \leqslant k \leqslant r \text{ and } c_k = c\}$.

Proof. Let us fix $c \in \{1, \ldots, m\}$. We recall that δ_c is the smallest exponent such that the row $\mathbf{E}\mathbf{J}^{\delta_c}{}_{c,*} = \mathcal{K}_{\mathbf{s}}(\mathbf{E})_{\phi_{\mathbf{s}}(c,\delta_c),*}$ is a linear combination of the rows in $\{\mathcal{K}_{\mathbf{s}}(\mathbf{E})_{i,*}, i < \phi_{\mathbf{s}}(c, \delta_c)\}$.

First, we assume that $\delta_c > 0$ and we let $d < \delta_c$. By definition of δ_c, the row at index $\phi_{\mathbf{s}}(c, d)$ in $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ is linearly independent from the rows with smaller indices. Thus, by minimality of the row rank profile, $\phi_{\mathbf{s}}(c, d) \in \{i_1, \ldots, i_r\}$.

In particular, choosing $d = 0$, we obtain that $\phi_{\mathbf{s}}(c, 0) \in \{i_1, \ldots, i_r\}$, or in other words, there is some $k \in \{1, \ldots, r\}$ such that $(c, 0) = (c_k, d_k)$. This proves that if $\delta_c > 0$, then $c_k = c$ for some $k \in \{1, \ldots, r\}$.

Now, we assume that $\delta_c = 0$ and we show that $c \neq c_k$ for all $k \in \{1, \ldots, r\}$. The definition of $\delta_c = 0$ and the third item in Lemma 7.14 together prove that for every $d \in \{0, \ldots, \delta\}$, the row $\mathbf{E}_{c,*}\mathbf{J}^d$ which is at index $\phi_{\mathbf{s}}(c, d)$ in $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ is a linear combination of the rows with smaller indices. Then, by minimality of the row rank profile, $\phi_{\mathbf{s}}(c, d) \notin \{i_1, \ldots, i_r\}$ for all $d \in \{0, \ldots, \delta\}$, and thus in particular $(c, d_k) \neq (c_k, d_k)$ for all $k \in \{1, \ldots, r\}$.

Finally, we assume that $\delta_c > 0$ and we show that $\delta_c = 1 + \max\{d_k \mid 1 \leqslant k \leqslant r \text{ and } c_k = c\}$. Using the first item with $d = \delta_c - 1$, there exists $\bar{k} \in \{1, \ldots, r\}$ such that $(c_{\bar{k}}, d_{\bar{k}}) = (c, \delta_c - 1)$. As in the previous paragraph, the definition of δ_c, the third item in Lemma 7.14, and the minimality of the row rank profile imply that $\phi_{\mathbf{s}}(c, d) \notin \{i_1, \ldots, i_r\}$ for all $d \in \{\delta_c, \ldots, \delta\}$; in particular, $(c, d_k) \neq (c_k, d_k)$ for all $k \in \{1, \ldots, r\}$ such that $d_k > d_{\bar{k}}$. Thus, we have $d_{\bar{k}} = \max\{d_k \mid 1 \leqslant k \leqslant r \text{ and } c_k = c\}$. □

Example 7.16 *(Minimal degree and row rank profile).* In the context of Examples 7.2 and 7.4,

- for the uniform shift, the row rank profile of $\mathcal{K}_{\mathbf{0}}(\mathbf{E})$ is $(0, 1, 3)$ with $0 = \phi_{\mathbf{0}}(0, 0)$, $1 = \phi_{\mathbf{0}}(1, 0)$, and $3 = \phi_{\mathbf{0}}(0, 1)$: then, the $\mathbf{0}$-minimal degree of $(\mathbf{E}, \mathbf{Z})$ is $(2, 1, 0)$;
- for the shift $\mathbf{s} = (0, 3, 6)$, the row rank profile of $\mathcal{K}(\mathbf{E})$ is $(0, 1, 2)$ with $0 = \phi_{\mathbf{s}}(0, 0)$, $1 = \phi_{\mathbf{s}}(0, 1)$, and $2 = \phi_{\mathbf{s}}(0, 2)$: the $\mathbf{s}$-minimal degree of $(\mathbf{E}, \mathbf{Z})$ is $(3, 0, 0)$;
- for the shift $\mathbf{t} = (3, 0, 2)$, the row rank profile of $\mathcal{K}_{\mathbf{t}}(\mathbf{E})$ is $(0, 1, 2)$ with $0 = \phi_{\mathbf{t}}(1, 0)$, $1 = \phi_{\mathbf{t}}(1, 1)$, and $2 = \phi_{\mathbf{t}}(1, 2)$: the $\mathbf{t}$-minimal degree of $(\mathbf{E}, \mathbf{Z})$ is $(0, 3, 0)$. □

In particular, the previous lemma implies a bound on the $\mathbf{s}$-minimal degree of $(\mathbf{E}, \mathbf{J})$. Since the minimal polynomial of the matrix $\mathbf{J} \in \mathbb{K}^{\sigma\times\sigma}$ is of degree at most δ, we have $\delta_c \leqslant \delta \leqslant \sigma$ for $1 \leqslant c \leqslant m$: we actually have the following stronger identity, which shows that the sum of $\delta_1, \ldots, \delta_m$ is at most σ.

Lemma 7.17. *Let* $\mathbf{J} \in \mathbb{K}^{\sigma\times\sigma}$ *and* $\mathbf{E} \in \mathbb{K}^{m\times\sigma}$, *and let* $\mathbf{s} \in \mathbb{N}^m$. *Let* $(\delta_1, \ldots, \delta_m)$ *be the* $\mathbf{s}$*-minimal degree of* $(\mathbf{E}, \mathbf{J})$. *Then,* $\delta_1 + \cdots + \delta_m = \mathrm{rank}(\mathcal{K}_{\mathbf{s}}(\mathbf{E}))$.

Proof. From Lemma 7.15, we see that one can partition the set $\{i_1, \ldots, i_r\}$ as the disjoint union of the sets $\{\phi_{\mathbf{s}}(c, d), 0 \leqslant d < \delta_c\}$ for each c with $\delta_c > 0$. This union has cardinality $\delta_1 + \cdots + \delta_m$, and the set $\{i_1, \ldots, i_r\}$ has cardinality $r = \mathrm{rank}(\mathcal{K}_{\mathbf{s}}(\mathbf{E}))$. □

Remark 7.18. Combining this with Lemma 7.12, one can directly deduce the following bound on the average row degree of minimal interpolation bases:

Let $\mathbf{J} \in \mathbb{K}^{\sigma\times\sigma}$ *and* $\mathbf{E} \in \mathbb{K}^{m\times\sigma}$, *and let* $\mathbf{s} \in \mathbb{N}^m$. *Then, for any* $\mathbf{s}$*-minimal interpolation basis* $\mathbf{P}$ *for* $(\mathbf{E}, \mathbf{J})$, *the sum of the* $\mathbf{s}$*-row degrees of* $\mathbf{P}$ *satisfies* $|\mathrm{rdeg}_{\mathbf{s}}(\mathbf{P})| \leqslant \sigma + |\mathbf{s}|$.

This bound has already been given before by Beckermann and Labahn (2000, Theorem 7.3.(b)), and also in the context of M-Padé approximation (Van Barel and Bultheel, 1992, Theorem 4.1), which includes order basis computation. This result is central regarding the cost of algorithms which compute shifted minimal interpolation bases since it gives a bound on the size of the output matrix. In particular it is a keystone for the efficiency of our divide-and-conquer algorithm in Section 3, where it gives a bound on the average row degree of all intermediate bases and thus allows fast computation of the product of bases (Section 4), of the change of shift (Section 5), and of the residuals (Section 6). □

Now, we show how to compute the row rank profile of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ efficiently. In the style of the algorithm of Keller-Gehrig (1985, p. 313), our algorithm processes submatrices of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ containing all rows up to some degree, doubling this degree at each iteration. The structure property in Lemma 7.14 allows us to always consider at most $2r$ rows of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$, discarding most rows with indices not in $\{i_1, \ldots, i_r\}$ without computing them. (There is one exception at the beginning, where the m rows of $\mathbf{E}$ are considered, with possibly m much larger than r.) This algorithm also returns the submatrix formed by the rows corresponding to the row rank profile, as well as the column rank profile of this submatrix, since they will both be useful later in Subsection 7.4.

Proposition 7.19. *Algorithm 8 is correct and uses $\mathcal{O}(\sigma^{\omega}(\lceil m/\sigma \rceil + \log(\delta)))$ operations in $\mathbb{K}$ if $\omega > 2$, and $\mathcal{O}(\sigma^2(\lceil m/\sigma \rceil + \log(\delta))\log(\sigma))$ operations if $\omega = 2$.*

Proof. The algorithm takes as input δ a power of 2: one can always ensure this by taking the next power of 2 without impacting the cost bound. After Steps **2**, **3**, and **4**, $(i_1, \ldots, i_r)$ correspond to the indices in $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ of the row rank profile of $\mathbf{E}$, and $\mathbf{M}$ is the submatrix of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ formed by the rows with indices in $\{i_1, \ldots, i_r\}$. Relying on the algorithm of Storjohann (2000, Section 2.2), Step **2** can be performed using $\mathcal{O}(m\sigma(\min(m,\sigma)^{\omega-2} + \log(\min(m,\sigma)))$ operations, and Step **6** can be computed using $\mathcal{O}(r\sigma(r^{\omega-2} + \log(r)))$ operations (where the logarithmic terms account for the possibility that $\omega = 2$). The loop performs $\log(\delta)$ iterations. In each iteration ℓ, since the matrix $\mathbf{M}$ has σ columns and has at most $2r$ rows with $r \leqslant \sigma$, one can compute the square $\mathbf{J}^{2^\ell}$ and the product $\mathbf{M}\mathbf{J}^{2^\ell}$ using $\mathcal{O}(\sigma^\omega)$ operations, and the row rank profile of $\mathbf{M}$ using $\mathcal{O}(r\sigma(r^{\omega-2} + \log(r)))$ operations (Storjohann, 2000, Section 2.2). Thus, overall, the for loop uses $\mathcal{O}(\sigma^\omega \log(\delta))$ operations if $\omega > 2$, and $\mathcal{O}(\sigma^2 \log(\delta)\log(\sigma))$ operations if $\omega = 2$. Adding these costs leads to the announced bound.

Let us now prove the correctness of the algorithm. For each $\ell \in \{0, \ldots, \log(\delta)\}$ let $\mathcal{I}_\ell = \{\phi_{\mathbf{s}}(c,d), 1 \leqslant c \leqslant m, 0 \leqslant d < 2^\ell\}$ denote the set of indices of rows of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ which correspond to degrees less than 2^ℓ, and let $\mathbf{K}_\ell$ be the submatrix of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ formed by the rows with indices in $\mathcal{I}_\ell$, that is, the submatrix $\mathbf{K}_\ell$ of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ which is a row permutation of the matrix

$$\begin{bmatrix} \mathbf{E} \\ \hline \mathbf{E}\mathbf{J} \\ \hline \vdots \\ \hline \mathbf{E}\mathbf{J}^{2^\ell - 1} \end{bmatrix} \in \mathbb{K}^{(2^\ell - 1)m \times \sigma};$$

for ease of presentation, we continue to index the rows of $\mathbf{K}_\ell$ with $\mathcal{I}_\ell$. Now, suppose that at the beginning of the iteration ℓ of the loop, $(i_1, \ldots, i_r)$ is the row rank profile of $\mathbf{K}_\ell$, and $\mathbf{M}$ is the submatrix of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ formed by the rows with indices in $\{i_1, \ldots, i_r\}$. Then, we claim that at the end of this iteration, $(k_{m_1}, \ldots, k_{m_r})$ is the row rank profile of $\mathbf{K}_{\ell+1}$; it is then obvious that the updated matrix $\mathbf{M}$, after Step **5.g**, is the corresponding submatrix of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$.

First, the indices $(i_{r+1}, \ldots, i_{2r})$ computed at Step **b** are in $\mathcal{I}_{\ell+1} - \mathcal{I}_\ell$, which is the set of indices of the rows of $\mathbf{K}_\ell\mathbf{J}^{2^\ell}$ in the matrix $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ (or in the matrix $\mathbf{K}_{\ell+1}$ since we chose to keep the same indexing). From Lemma 7.14, we know that if two indices $i = \phi_{\mathbf{s}}(c,d) < i' = \phi_{\mathbf{s}}(c',d')$ are in $\mathcal{I}_\ell$, then we also have $\phi_{\mathbf{s}}(c, d+2^\ell) < \phi_{\mathbf{s}}(c', d'+2^\ell)$ in $\mathcal{I}_{\ell+1} - \mathcal{I}_\ell$. This means that $\mathbf{K}_\ell\mathbf{J}^{2^\ell}$ is not only formed by the rows of $\mathbf{K}_{\ell+1}$ with indices in $\mathcal{I}_{\ell+1} - \mathcal{I}_\ell$: it is actually the submatrix of $\mathbf{K}_{\ell+1}$ formed by these rows, keeping the same row order.

In particular, for a given $k \in \mathcal{I}_\ell$, if the row k of $\mathbf{K}_\ell$ is a linear combination of the rows of this matrix with smaller indices, then the same property holds in the matrix $\mathbf{K}_{\ell+1}$; and similarly if the row $k \in \mathcal{I}_{\ell+1} - \mathcal{I}_\ell$ of $\mathbf{K}_\ell\mathbf{J}^{2^\ell}$ is a linear combination of the rows of this matrix with smaller indices, then the same holds in $\mathbf{K}_{\ell+1}$. Another consequence is that the sequence $(i_{r+1},\ldots,i_{2r})$ defined in Step **5.b** is strictly increasing, as stated in Step **5.c**; besides, it does not share any common element with $(i_1,\ldots,i_r)$, so that their merge $(k_1,\ldots,k_{2r})$ in Step **5.c** is unique and strictly increasing.

Now, since the row rank profile of $\mathbf{K}_\ell\mathbf{J}^{2^\ell}$ is a subsequence of the row rank profile of $\mathbf{K}_\ell$, the row rank profile of the submatrix of $\mathbf{K}_{\ell+1}$ formed by the rows in $\mathcal{I}_{\ell+1}-\mathcal{I}_\ell$ is a subsequence of $(i_{r+1},\ldots,i_{2r})$. Thus, if k is an index in $\mathcal{I}_{\ell+1}-\{k_1,\ldots,k_{2r}\}$, then the row k of $\mathbf{K}_{\ell+1}$ is a linear combination of the rows with smaller indices, and thus k will not appear in the row rank profile of $\mathbf{K}_{\ell+1}$. Thus, the row rank profile of $\mathbf{K}_{\ell+1}$, that is, of the submatrix of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ formed by the rows in $\mathcal{I}_{\ell+1}$, is a subsequence of $(k_1,\ldots,k_{2r})$. This justifies that in Steps **5.e** and **5.f** one may only focus on the rows with indices in $\{k_1,\ldots,k_{2r}\}$. The conclusion follows. □

Algorithm 8 *(*KrylovRankProfile*).*

Input:

- matrix $\mathbf{E}\in\mathbb{K}^{m\times\sigma}$,
- matrix $\mathbf{J}\in\mathbb{K}^{\sigma\times\sigma}$,
- shift $\mathbf{s}\in\mathbb{N}^m$,
- a bound δ on the degree of the minimal polynomial of $\mathbf{E}$, with δ a power of 2 in $\{1,\ldots,2\sigma-1\}$.

Output:

- the row rank profile $(i_1,\ldots,i_r)$ of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$,
- the submatrix $\mathcal{P}_{\mathbf{s}}(\mathbf{E})$ of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ formed by the rows with indices in $\{i_1,\ldots,i_r\}$,
- the column rank profile $(j_1,\ldots,j_r)$ of $\mathcal{P}_{\mathbf{s}}(\mathbf{E})$.

1. compute $\phi_{\mathbf{s}}$ as in Definition 7.3
2. $r,(c_1,\ldots,c_r)\leftarrow$ RowRankProfile($\mathbf{E}$)
3. $(i_1,\ldots,i_r)\leftarrow(\phi_{\mathbf{s}}(c_1,0),\ldots,\phi_{\mathbf{s}}(c_r,0))$
4. $\mathbf{M}\leftarrow$ submatrix of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ with rows of indices in $\{i_1,\ldots,i_r\}$
5. For ℓ from 0 to $\log(\delta)$,
  a. $(c_1,d_1)\leftarrow\phi_{\mathbf{s}}^{-1}(i_1),\ \ldots,\ (c_r,d_r)\leftarrow\phi_{\mathbf{s}}^{-1}(i_r)$
  b. $i_{r+1}\leftarrow\phi_{\mathbf{s}}(c_1,d_1+2^\ell),\ \ldots,\ i_{2r}\leftarrow\phi_{\mathbf{s}}(c_r,d_r+2^\ell)$
  c. $(k_1,\ldots,k_{2r})\leftarrow$ merge the increasing sequences $(i_1,\ldots,i_r)$ and $(i_{r+1},\ldots,i_{2r})$
  d. compute $\mathbf{M}\mathbf{J}^{2^\ell}\in\mathbb{K}^{r\times\sigma}$, the rows at indices $i_{r+1},\ldots,i_{2r}$ in $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$
  e. $\mathbf{M}\leftarrow$ the submatrix of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ formed by the rows in $\mathbf{M}$ and $\mathbf{M}\mathbf{J}^{2^\ell}$; that is, the rows of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ with indices in $\{k_1,\ldots,k_{2r}\}$
  f. $r',(m_1,\ldots,m_{r'})\leftarrow$ RowRankProfile($\mathbf{M}$)
  g. $\mathbf{M}\leftarrow$ the submatrix of $\mathbf{M}$ formed by rows with indices in $\{m_1,\ldots,m_{r'}\}$
  h. $r,(i_1,\ldots,i_r)\leftarrow r',(k_{m_1},\ldots,k_{m_{r'}})$
6. $(j_1,\ldots,j_r)\leftarrow$ ColRankProfile($\mathbf{M}$)
7. Return $(i_1,\ldots,i_r)$, $\mathbf{M}$, and $(j_1,\ldots,j_r)$.

7.4. Computing minimal interpolation bases via linearization

As noted in Remark 7.13, an $\mathbf{s}$-minimal interpolation basis for $(\mathbf{E},\mathbf{J})$ can be retrieved from linear relations which express the rows of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ of indices $\{\phi_{\mathbf{s}}(1,\delta_1),\ldots,\phi_{\mathbf{s}}(m,\delta_m)\}$ as combinations of the rows with smaller indices. Concerning the latter rows, one can for example restrict to those given by the row rank profile $(i_1,\ldots,i_r)$: thus, one can build an interpolation basis by considering only $r+m\leqslant\sigma+m$ rows in $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$. In many useful cases, $\sigma+m$ is significantly smaller than the total number of rows $m(\delta+1)$ in the matrix.

Definition 7.20. Let $\mathbf{J}\in\mathbb{K}^{\sigma\times\sigma}$ and $\mathbf{E}\in\mathbb{K}^{m\times\sigma}$, and let $\mathbf{s}\in\mathbb{N}^m$. Then $\mathcal{P}_{\mathbf{s}}(\mathbf{E})\in\mathbb{K}^{r\times\sigma}$ is the submatrix of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ formed by its rows with indices in $\{i_1,\ldots,i_r\}$, and $\mathcal{T}_{\mathbf{s}}(\mathbf{E})\in\mathbb{K}^{m\times\sigma}$ is the submatrix of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ formed by the rows with indices in $\{\phi_{\mathbf{s}}(1,\delta_1),\ldots,\phi_{\mathbf{s}}(m,\delta_m)\}$.

The matrix $\mathcal{P}_{\mathbf{s}}(\mathbf{E}) \in \mathbb{K}^{r\times\sigma}$ can be thought of as a *pivot* matrix, since its rows are used as pivots to find relations through the elimination of the rows in $\mathcal{T}_{\mathbf{s}}(\mathbf{E}) \in \mathbb{K}^{m\times\sigma}$, which we therefore think of as the *target* matrix. From Subsection 7.2, we know that these relations correspond to an interpolation basis $\mathbf{P}$ in $\mathbf{s}$-weak Popov form. It turns out that restricting our view of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ to the submatrix $\mathcal{P}_{\mathbf{s}}(\mathbf{E})$ leads to find such relations with a minimal number of coefficients, which corresponds to a stronger type of minimality: $\mathbf{P}$ is in $\mathbf{s}$-Popov form (Kailath, 1980; Beckermann et al., 2006).

Definition 7.21 *(shifted Popov form).* Let $\mathbf{P} \in \mathbb{K}[X]^{m\times m}$ have full rank, and let $\mathbf{s}$ in $\mathbb{N}^m$. Then, $\mathbf{P}$ is said to be in $\mathbf{s}$-*Popov form* if the $\mathbf{s}$-pivot entries are monic and on the diagonal of $\mathbf{P}$, and in each column of $\mathbf{P}$ the nonpivot entries have degree less than the pivot entry.

A matrix in $\mathbf{s}$-Popov form is in particular in $\mathbf{s}$-weak Popov form and $\mathbf{s}$-reduced; besides, this is a normal form in the sense that, for a given $\mathbf{A} \in \mathbb{K}[X]^{m\times m}$ with full rank and a given shift $\mathbf{s}$, there is a unique matrix $\mathbf{P}$ in $\mathbf{s}$-Popov form which is unimodularly equivalent to $\mathbf{A}$. In particular, given $(\mathbf{E}, \mathbf{J})$, for each shift $\mathbf{s}$ there is a unique ($\mathbf{s}$-minimal) interpolation basis for $(\mathbf{E}, \mathbf{J})$ which is in $\mathbf{s}$-Popov form.

Since all rows in $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ are linear combinations of those in the submatrix $\mathcal{P}_{\mathbf{s}}(\mathbf{E})$, there is an $m \times r$ matrix $\mathcal{R}_{\mathbf{s}}(\mathbf{E})$ such that $\mathcal{T}_{\mathbf{s}}(\mathbf{E}) = \mathcal{R}_{\mathbf{s}}(\mathbf{E})\mathcal{P}_{\mathbf{s}}(\mathbf{E})$, which we think of as the *relation* matrix; besides, since the pivot matrix has full rank, this defines $\mathcal{R}_{\mathbf{s}}(\mathbf{E})$ uniquely. Then, the linear relations that we are looking for are $[-\mathcal{R}_{\mathbf{s}}(\mathbf{E})|\mathbf{I}_m]$, and they can be computed for example using Gaussian elimination on the rows of $[\mathcal{P}_{\mathbf{s}}(\mathbf{E})^{\mathsf{T}}|\mathcal{T}_{\mathbf{s}}(\mathbf{E})^{\mathsf{T}}]^{\mathsf{T}}$. More precisely, $[-\mathcal{R}_{\mathbf{s}}(\mathbf{E})|\mathbf{I}_m]$ is the set of columns with indices in $(i_1, \ldots, i_r, \phi_{\mathbf{s}}(1, \delta_1), \ldots, \phi_{\mathbf{s}}(m, \delta_m))$ of the relations that we are looking for between the rows of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$, and the interpolation basis in $\mathbf{s}$-Popov form is the compression of these relations. Generally, given a matrix $\mathbf{A}$ in $\mathbb{K}^{n\times(r+m)}$ for some n, we see it as formed by the columns with indices $(i_1, \ldots, i_r, \phi_{\mathbf{s}}(1, \delta_1), \ldots, \phi_{\mathbf{s}}(m, \delta_m))$ (in this order) of a matrix $\mathbf{B}$ in $\mathbb{K}^{n\times m(\delta+1)}$ which has other columns zero. Then, the compression of $\mathbf{A}$ is the compression $\mathcal{C}_{\mathbf{s}}(\mathbf{B})$ of $\mathbf{B}$ as defined in Subsection 7.1; we abusively denote it $\mathcal{C}_{\mathbf{s}}(\mathbf{A})$ since there will be no ambiguity.

Lemma 7.22. *Let $\mathbf{J} \in \mathbb{K}^{\sigma\times\sigma}$ and $\mathbf{E} \in \mathbb{K}^{m\times\sigma}$, and let $\mathbf{s} \in \mathbb{N}^m$. Let $\mathcal{P}_{\mathbf{s}}(\mathbf{E}) \in \mathbb{K}^{r\times\sigma}$ and $\mathcal{T}_{\mathbf{s}}(\mathbf{E}) \in \mathbb{K}^{m\times\sigma}$ be as in Definition 7.20, and let $\mathcal{R}_{\mathbf{s}}(\mathbf{E})$ be the unique matrix in $\mathbb{K}^{m\times r}$ such that $\mathcal{T}_{\mathbf{s}}(\mathbf{E}) = \mathcal{R}_{\mathbf{s}}(\mathbf{E})\mathcal{P}_{\mathbf{s}}(\mathbf{E})$. Then,*

$$\mathbf{P} = \mathcal{C}_{\mathbf{s}}([-\mathcal{R}_{\mathbf{s}}(\mathbf{E})\,|\,\mathbf{I}_m])$$

is an interpolation basis for $(\mathbf{E}, \mathbf{J})$ in $\mathbf{s}$-Popov form.

Besides, if $(j_1, \ldots, j_r)$ denotes the column rank profile of $\mathcal{P}_{\mathbf{s}}(\mathbf{E})$, and $\mathbf{C} \in \mathbb{K}^{r\times r}$ and $\mathbf{D} \in \mathbb{K}^{m\times r}$ are the submatrices of $\mathcal{P}_{\mathbf{s}}(\mathbf{E})$ and $\mathcal{T}_{\mathbf{s}}(\mathbf{E})$, respectively, formed by the columns with indices in $\{j_1, \ldots, j_r\}$, then we have

$$\mathcal{R}_{\mathbf{s}}(\mathbf{E}) = \mathbf{D}\mathbf{C}^{-1}.$$

Proof. First, restricting the identity $\mathcal{T}_{\mathbf{s}}(\mathbf{E}) = \mathcal{R}_{\mathbf{s}}(\mathbf{E})\mathcal{P}_{\mathbf{s}}(\mathbf{E})$ to the submatrices with column indices in $\{j_1, \ldots, j_r\}$ we have in particular $\mathbf{D} = \mathcal{R}_{\mathbf{s}}(\mathbf{E})\mathbf{C}$. By construction, $\mathbf{C}$ is invertible and thus $\mathcal{R}_{\mathbf{s}}(\mathbf{E}) = \mathbf{D}\mathbf{C}^{-1}$.

Let $\mathbf{R} \in \mathbb{K}^{m\times m(\delta+1)}$ be the matrix whose columns at indices $i_1, \ldots, i_r$, $\phi_{\mathbf{s}}(1, \delta_1)$, $\ldots$, $\phi_{\mathbf{s}}(m, \delta_m)$ are the columns $1, \ldots, r+m$ of $[-\mathcal{R}_{\mathbf{s}}(\mathbf{E})|\mathbf{I}_m]$, respectively, and other columns are zero; let also $\mathbf{P} = \mathcal{C}_{\mathbf{s}}([-\mathcal{R}_{\mathbf{s}}(\mathbf{E})|\mathbf{I}_m]) = \mathcal{C}_{\mathbf{s}}(\mathbf{R})$. By construction, every row c of $\mathbf{P}$ is the compression $\mathcal{C}_{\mathbf{s}}(\mathbf{R}_{c,*})$ of a linear relation between the rows of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ and is thus an interpolant for $(\mathbf{E}, \mathbf{J})$. We will further prove that $\mathbf{P}$ is in $\mathbf{s}$-Popov form with $\mathbf{s}$-pivot degree $(\delta_1, \ldots, \delta_m)$; in particular, this implies that $\mathbf{P}$ is $\mathbf{s}$-reduced and has $\mathbf{s}$-row degree $(s_1 + \delta_1, \ldots, s_m + \delta_m)$, so that Lemma 7.12 shows that $\mathbf{P}$ is an interpolation basis for $(\mathbf{E}, \mathbf{J})$. For $k \in \{1, \ldots, r\}$, we write $i_k = \phi_{\mathbf{s}}(c_k, d_k)$ for some unique (c_k, d_k). We fix $c \in \{1, \ldots, m\}$.

First, we consider the column c of $\mathbf{P}$ and we show that all its entries have degree less than δ_c except the entry on the diagonal, which is monic and has degree exactly δ_c. Indeed, for any k such that $c_k = c$, by definition of $\mathcal{C}_{\mathbf{s}}(\mathbf{R})$ the column i_k of $\mathbf{R}$ is compressed into the coefficient of degree d_k in the column c of $\mathbf{P}$, and by Lemma 7.15 we know that $d_k < \delta_c$. Besides, the column of $\mathbf{R}$ at index

$\phi_{\mathbf{s}}(c, \delta_c)$, which has all its entries 0 except the entry on row c which is 1, only brings a coefficient 1 of degree δ_c in the diagonal entry (c, c) of $\mathbf{P}$.

Second, we consider the row c of $\mathbf{P}$ and we show that it has $\mathbf{s}$-pivot index c and $\mathbf{s}$-pivot degree δ_c. Thanks to Lemma 7.6, it is enough to show that the rightmost nonzero entry in the row c of $\mathbf{R}$ is the entry 1 at column index $\phi_{\mathbf{s}}(c, \delta_c)$. All entries in the row c of $\mathbf{R}$ with indices greater than $\phi_{\mathbf{s}}(c, \delta_c)$ and not in $\{i_1, \ldots, i_r\}$ are obviously zero. Now, by definition of δ_c, we know that the row of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ at index $\phi_{\mathbf{s}}(c, \delta_c)$ is a linear combination of the rows at indices smaller than $\phi_{\mathbf{s}}(c, \delta_c)$; in particular, because the rows of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ at indices $i_1, \ldots, i_r$ are linearly independent, the linear combination given by the row c of $\mathbf{R}$ has entries 0 on the columns at indices i_k for k such that $i_k > \phi_{\mathbf{s}}(c, \delta_c)$. □

Now, we turn to the fast computation of an interpolation basis $\mathbf{P}$ for $(\mathbf{E}, \mathbf{J})$ in $\mathbf{s}$-Popov form. In view of what precedes, this boils down to two steps, detailed in Algorithm 9: first, we compute the row rank profile $(i_1, \ldots, i_r)$ of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ from which we also deduce the $\mathbf{s}$-minimal degree $(\delta_1, \ldots, \delta_m)$, and second, we compute the linear relations $\mathbf{D}\mathbf{C}^{-1}$. We now prove Proposition 7.1, by showing that Algorithm 9 is correct and uses $\mathcal{O}(\sigma^\omega(\lceil m/\sigma\rceil + \log(\delta)))$ operations in $\mathbb{K}$ if $\omega > 2$, and $\mathcal{O}(\sigma^\omega(\lceil m/\sigma\rceil + \log(\delta))\log(\sigma))$ operations if $\omega = 2$.

Proof of Proposition 7.1. The correctness follows from Lemmas 7.15 and 7.22 and from the correctness of the algorithm KrylovRankProfile. Since $r \leqslant \sigma$, the computation of $\mathbf{C}^{-1}$ at Step **7** uses $\mathcal{O}(r^\omega) \subset \mathcal{O}(\sigma^\omega)$ operations, and the computation of $\mathbf{D}\mathbf{C}^{-1}$ uses $\mathcal{O}(\sigma^\omega)$ operations when $m \leqslant \sigma$, and $\mathcal{O}(m\sigma^{\omega-1})$ operations when $\sigma \leqslant m$. Then, the announced cost bound follows from Proposition 7.19. □

Algorithm 9 *(*LinearizationInterpolationBasis*).*

Input:

- matrix $\mathbf{E} \in \mathbb{K}^{m\times\sigma}$,
- matrix $\mathbf{J} \in \mathbb{K}^{\sigma\times\sigma}$,
- shift $\mathbf{s} \in \mathbb{N}^m$,
- a bound δ on the degree of the minimal polynomial of $\mathbf{J}$, with δ a power of 2 in $\{1, \ldots, 2\sigma - 1\}$.

Output: the interpolation basis $\mathbf{P} \in \mathbb{K}[X]^{m\times m}_{\leqslant\delta}$ for $(\mathbf{E}, \mathbf{J})$ in $\mathbf{s}$-Popov form.

1. compute $\psi_{\mathbf{s}}$ and $\phi_{\mathbf{s}}$ as in Definition 7.3
2. $(i_1, \ldots, i_r), \mathcal{P}_{\mathbf{s}}(\mathbf{E}), (j_1, \ldots, j_r) \leftarrow$ KrylovRankProfile$(\mathbf{E}, \mathbf{J}, \mathbf{s}, \delta)$
3. For $1 \leqslant k \leqslant r$, compute $(c_k, d_k) \leftarrow \phi_{\mathbf{s}}^{-1}(i_k)$
4. For $1 \leqslant c \leqslant m$, compute $\delta_c \leftarrow 1 + \max\{d_k \mid 1 \leqslant k \leqslant r \text{ and } c_k = c\}$ if the set is nonempty, and $\delta_c \leftarrow 0$ if it is empty
5. $\mathcal{T}_{\mathbf{s}}(\mathbf{E}) \leftarrow$ submatrix of $\mathcal{K}_{\mathbf{s}}(\mathbf{E})$ formed by the rows with indices in $\{\phi_{\mathbf{s}}(1, \delta_1), \ldots, \phi_{\mathbf{s}}(m, \delta_m)\}$
6. $\mathbf{C}, \mathbf{D} \leftarrow$ submatrices of $\mathcal{P}_{\mathbf{s}}(\mathbf{E})$ and $\mathcal{T}_{\mathbf{s}}(\mathbf{E})$, respectively, formed by the columns with indices in $\{j_1, \ldots, j_r\}$
7. compute $\mathcal{R}_{\mathbf{s}}(\mathbf{E}) \leftarrow \mathbf{D}\mathbf{C}^{-1}$
8. $\mathbf{P} \leftarrow \mathcal{C}_{\mathbf{s}}([-\mathcal{R}_{\mathbf{s}}(\mathbf{E}) \mid \mathbf{I}_m])$
9. Return $\mathbf{P}$

Acknowledgements

We thank B. Beckermann, E. Kaltofen, G. Labahn, and E. Tsigaridas for useful discussions. We also thank the reviewers for their thorough reading and helpful comments. C.-P. Jeannerod and G. Villard were partly supported by the ANR HPAC project (ANR 11 BS02 013). V. Neiger was supported by the international mobility grants from *Projet Avenir Lyon Saint-Étienne*, *Mitacs Globalink – Inria*, and Explo'ra Doc from *Région Rhône-Alpes*. É. Schost was supported by NSERC and by the Canada Research Chairs program.

Appendix A. Bounds for polynomial matrix multiplication functions

In this appendix, we give upper bounds for the quantities in Definition 3.3.

Lemma A.1. *We have the upper bounds*

$$\begin{aligned} \mathsf{MM}'(m,d) &\in \mathcal{O}(m^{\omega-1}\mathsf{M}(md)) && \text{if } \omega > 2, \\ \mathsf{MM}'(m,d) &\in \mathcal{O}(m\mathsf{M}(md)\log(m)) && \text{if } \omega = 2, \\ \mathsf{MM}''(m,d) &\in \mathcal{O}(m^{\omega}\mathsf{M}(d)\log(d)). \end{aligned}$$

Proof. It is enough to show these bounds for m and d powers of 2. The bound on $\mathsf{MM}''(m,d)$ follows from the super-linearity property $2^j\mathsf{M}(2^{-j}d) \leqslant \mathsf{M}(d)$.

Using the super-linearity property $\mathsf{M}(2^{-i}md) \leqslant 2^{-i}\mathsf{M}(md)$, we obtain $\mathsf{MM}'(m,d) = \sum_{0\leqslant i\leqslant \log(m)} 2^{-i}m\mathsf{MM}(2^i, 2^{-i}md) \in \mathcal{O}(\sum_{0\leqslant i\leqslant\log(m)} 2^{i(\omega-2)}m\mathsf{M}(md))$. This concludes the proof since we have $\sum_{0\leqslant i\leqslant\log(m)} 2^{i(\omega-2)} \in \Theta(m^{\omega-2}+\log(m))$, where the logarithmic term accounts for the possibility that $\omega = 2$. □

Lemma A.2. *We have the upper bounds*

$$\begin{aligned} \overline{\mathsf{MM}'}(m,d) &\in \mathcal{O}(m^{\omega-1}\mathsf{M}(md)) && \text{if } \omega > 2, \\ \overline{\mathsf{MM}'}(m,d) &\in \mathcal{O}(m\mathsf{M}(md)\log(m)^2) && \text{if } \omega = 2, \\ \overline{\mathsf{MM}''}(m,d) &\in \mathcal{O}(m^{\omega-1}\mathsf{M}(md) + m^{\omega}\mathsf{M}(d)\log(d)) && \text{if } \omega > 2, \\ \overline{\mathsf{MM}''}(m,d) &\in \mathcal{O}(m\mathsf{M}(md)\log(m)^2 + m^2\mathsf{M}(d)\log(d)\log(m)) && \text{if } \omega = 2. \end{aligned}$$

Proof. It is enough to show these bounds for m and d powers of 2. The first two bounds are obtained from Lemma A.1, which implies that

$$\sum_{i=0}^{\log(m)} 2^i\mathsf{MM}'(2^{-i}m, 2^id) \in \mathcal{O}\left(\left(\sum_{i=0}^{\log(m)} 2^{i(2-\omega)}m^{\omega-1}\mathsf{M}(md)\right) + m\mathsf{M}(md)\log(m)^2\right),$$

and from the fact that $\sum_{0\leqslant i\leqslant\log(m)} 2^{i(2-\omega)}$ is upper bounded by a constant if $\omega > 2$.

Now, we focus on the last two bounds. By definition,

$$\overline{\mathsf{MM}''}(m,d) \in \mathcal{O}\left(\sum_{i=0}^{\log(m)} 2^i \sum_{j=0}^{\log(2^id)} 2^j(2^{-i}m)^{\omega}\mathsf{M}(2^{i-j}d)\right).$$

In the inner sum, j goes from 0 to $\log(2^id) = i + \log(d)$: we will separately study the first terms with $j \leqslant i$ and the remaining terms with $j > i$.

First, using the super-linearity property $\mathsf{M}(2^{i-j}d) \leqslant 2^{i-j}\mathsf{M}(md)/m$, we obtain

$$\begin{aligned} \sum_{i=0}^{\log(m)} 2^i \sum_{j=0}^{i} 2^j(2^{-i}m)^{\omega}\mathsf{M}(2^{i-j}d) &\in \mathcal{O}\left(\sum_{i=0}^{\log(m)} (i+1)2^{i(2-\omega)}m^{\omega-1}\mathsf{M}(md)\right) \\ &\in \mathcal{O}(m^{\omega-1}\mathsf{M}(md) + m\mathsf{M}(md)\log(m)^2), \end{aligned}$$

since when $\omega > 2$, the sum $\sum_{0\leqslant i\leqslant\log(m)}(i+1)2^{i(2-\omega)}$ is known to be less than its limit $(1-2^{2-\omega})^{-2}$ when $m \to \infty$. We note that the second term $m\mathsf{M}(md)\log(m)^2$ accounts for the possibility that $\omega = 2$.

Then, using the super-linearity property $\mathsf{M}(2^{i-j}d) \leqslant 2^{i-j}\mathsf{M}(d)$ when $j > i$, we obtain

$$\sum_{i=0}^{\log(m)} 2^i \sum_{j=i+1}^{i+\log(d)} 2^j (2^{-i}m)^{\omega} \mathsf{M}(2^{i-j}d) \leqslant \sum_{i=0}^{\log(m)} 2^{i(2-\omega)} m^{\omega} \mathsf{M}(d) \log(d)$$
$$\in \mathcal{O}(m^{\omega}\mathsf{M}(d)\log(d) + m^2\mathsf{M}(d)\log(d)\log(m)),$$

which concludes the proof. □

Lemma A.3. *Let $\bar{d}$ denote the power of 2 such that $d \leqslant \bar{d} < 2d$; we have the upper bounds*

$$\textstyle\sum_{0\leqslant i\leqslant \log(\bar{d})} 2^i \overline{\mathsf{MM}'}(m, 2^{-i}\bar{d}) \in \mathcal{O}(m^{\omega-1}\mathsf{M}(md) + m^{\omega}\mathsf{M}(d)\log(d)) \qquad \text{if } \omega > 2,$$
$$\textstyle\sum_{0\leqslant i\leqslant \log(\bar{d})} 2^i \overline{\mathsf{MM}'}(m, 2^{-i}\bar{d}) \in \mathcal{O}(m\mathsf{M}(md)\log(m)^3 + m^2\mathsf{M}(d)\log(d)\log(m)^2) \qquad \text{if } \omega = 2,$$
$$\textstyle\sum_{0\leqslant i\leqslant \log(\bar{d})} 2^i \overline{\mathsf{MM}''}(m, 2^{-i}\bar{d}) \in \mathcal{O}(m^{\omega-1}\mathsf{M}(md) + m^{\omega}\mathsf{M}(d)\log(d)^2) \qquad \text{if } \omega > 2,$$
$$\textstyle\sum_{0\leqslant i\leqslant \log(\bar{d})} 2^i \overline{\mathsf{MM}''}(m, 2^{-i}\bar{d}) \in \mathcal{O}(m\mathsf{M}(md)\log(m)^3 + m^2\mathsf{M}(d)\log(d)^2\log(m)) \qquad \text{if } \omega = 2.$$

Proof. It is enough to show these bounds for m and d powers of 2; in particular, $d = \bar{d}$.

Let us study the first two bounds. By definition,

$$\sum_{i=0}^{\log(d)} 2^i \overline{\mathsf{MM}'}(m, 2^{-i}d) = \sum_{i=0}^{\log(d)} 2^i \sum_{j=0}^{\log(m)} 2^j \sum_{k=0}^{\log(m)-j} 2^k \mathsf{MM}(2^{-j-k}m, 2^{j+k-i}d)$$
$$= \sum_{j=0}^{\log(m)} \sum_{k=0}^{\log(m)-j} \sum_{i=0}^{\log(d)} 2^{i+j+k} \mathsf{MM}(2^{-j-k}m, 2^{j+k-i}d).$$

Considering the terms with $i \leqslant j+k$, we use $\mathsf{M}(2^{j+k-i}d) \leqslant 2^{j+k-i}\mathsf{M}(md)/m$ to obtain

$$\sum_{j=0}^{\log(m)} \sum_{k=0}^{\log(m)-j} \sum_{i=0}^{j+k} 2^{i+j+k} \mathsf{MM}(2^{-j-k}m, 2^{j+k-i}d)$$
$$\in \mathcal{O}\left(\sum_{j=0}^{\log(m)} \sum_{k=0}^{\log(m)-j} (j+k+1) 2^{(j+k)(2-\omega)} m^{\omega-1}\mathsf{M}(md)\right),$$

from which we conclude since the sum $\sum_{0\leqslant j\leqslant \log(m)} \sum_{0\leqslant k\leqslant \log(m)-j} (j+k+1)2^{(j+k)(2-\omega)}$ is $\mathcal{O}(1)$ if $\omega > 2$, and $\mathcal{O}(\log(m)^3)$ if $\omega = 2$.

Now, considering the terms with $i > j+k$, we use $\mathsf{M}(2^{j+k-i}d) \leqslant 2^{j+k-i}\mathsf{M}(d)$ to obtain

$$\sum_{j=0}^{\log(m)} \sum_{k=0}^{\log(m)-j} \sum_{i=j+k+1}^{\log(d)} 2^{i+j+k} \mathsf{MM}(2^{-j-k}m, 2^{j+k-i}d)$$
$$\in \mathcal{O}\left(\sum_{j=0}^{\log(m)} \sum_{k=0}^{\log(m)-j} 2^{(j+k)(2-\omega)} m^{\omega}\mathsf{M}(d)\log(d)\right),$$

where again the sum on j and k is $\mathcal{O}(1)$ if $\omega > 2$, and $\mathcal{O}(\log(m)^2)$ if $\omega = 2$.

Then, we study the last two bounds. By definition,

$$\sum_{i=0}^{\log(d)} 2^i \overline{\mathsf{MM}''}(m, 2^{-i}d) = \sum_{i=0}^{\log(d)} 2^i \sum_{j=0}^{\log(m)} 2^j \sum_{k=0}^{\log(d)+j-i} 2^k \mathsf{MM}(2^{-j}m, 2^{j-i-k}d)$$

$$= \sum_{j=0}^{\log(m)} \sum_{i=0}^{\log(d)} \sum_{k=0}^{\log(d)+j-i} 2^{i+j+k}\mathsf{MM}(2^{-j}m, 2^{j-i-k}d).$$

Considering the terms with $k > j-i$, we use $\mathsf{M}(2^{j-i-k}d) \leqslant 2^{j-i-k}\mathsf{M}(d)$ to obtain

$$\sum_{j=0}^{\log(m)} \sum_{i=0}^{\log(d)} \sum_{k=\max(0,1+j-i)}^{\log(d)+j-i} 2^{i+j+k}\mathsf{MM}(2^{-j}m, 2^{j-i-k}d) \in \mathcal{O}\left(\sum_{j=0}^{\log(m)} 2^{j(2-\omega)}m^{\omega}\mathsf{M}(d)\log(d)^2\right);$$

this is $\mathcal{O}(m^{\omega}\mathsf{M}(d)\log(d)^2)$ if $\omega > 2$ and $\mathcal{O}(m\mathsf{M}(d)\log(d)^2\log(m))$ if $\omega = 2$.

Now, considering the terms with $0 \leqslant k \leqslant j-i$, and thus also $i \leqslant j$, we use $\mathsf{M}(2^{j-i-k}d) \leqslant 2^{j-i-k}\mathsf{M}(md)/m$ to obtain

$$\sum_{j=0}^{\log(m)} \sum_{i=0}^{j} \sum_{k=0}^{j-i} 2^{i+j+k}\mathsf{MM}(2^{-j}m, 2^{j-i-k}d) \in \mathcal{O}\left(\sum_{j=0}^{\log(m)} (j+1)^2 2^{j(2-\omega)}m^{\omega-1}\mathsf{M}(md)\right).$$

This gives the conclusion, since $\sum_{j=0}^{\log(m)}(j+1)^2 2^{j(2-\omega)}$ is $\mathcal{O}(1)$ if $\omega > 2$, and $\mathcal{O}(\log(m)^3)$ if $\omega = 2$. □

Appendix B. Cost analysis for the computation of minimal nullspace bases

Here, we give a detailed cost analysis for the minimal nullspace basis algorithm of Zhou et al. (2012), which we rewrite in Algorithm 10 using our convention here that basis vectors are rows of the basis matrix (whereas in the above reference they are its columns). Furthermore, we assume that the input matrix has full rank, which allows us to better control the dimensions of the matrices encountered in the computations: in the recursive calls, we always have input matrices with more rows than columns.

Here, the quantity $\mathsf{MM}''(m,d) = \sum_{0\leqslant j\leqslant \log(d)} 2^j\mathsf{MM}(m, 2^{-j}d)$ arises in the cost analysis of fast algorithms for the computation of Hermite–Padé approximants (Beckermann and Labahn, 1994; Giorgi et al., 2003), which use a divide-and-conquer approach on the degree d. The minimal nullspace basis algorithm in Zhou et al. (2012) follows a divide-and-conquer approach on the dimension of the input matrix, and computes at each node of the recursion some products of matrices with unbalanced row degrees as well as a minimal basis of Hermite–Padé approximants. In particular, its cost will be expressed using the quantities $\overline{\mathsf{MM}}'(m,d)$ and $\overline{\mathsf{MM}}''(m,d)$ introduced in Definition 3.3.

The following result refines the cost analysis in Zhou et al. (2012, Theorem 4.1), counting the logarithmic factors.

Proposition B.1. *Let* $\mathbf{F}$ *in* $\mathbb{K}[X]^{m\times n}$ *have full rank with* $m \geqslant n$*, and let* $\mathbf{s}$ *in* $\mathbb{N}^m$ *which bounds the row degree of* $\mathbf{F}$ *componentwise. Let* $\xi \geqslant m$ *be an integer such that* $|\mathbf{s}| \leqslant \xi$*. Assuming that* $m \in \mathcal{O}(n)$*, Algorithm 10 computes an* $\mathbf{s}$*-minimal nullspace basis of* $\mathbf{F}$ *using*

$$\begin{aligned} &\mathcal{O}(\overline{\mathsf{MM}}'(m,\xi/m) + \overline{\mathsf{MM}}''(m,\xi/m)) \\ &\quad\subseteq \mathcal{O}(m^{\omega-1}\mathsf{M}(\xi) + m^{\omega}\mathsf{M}(\xi/m)\log(\xi/m)) && \text{if } \omega > 2, \\ &\quad\subseteq \mathcal{O}(m\mathsf{M}(\xi)\log(m)^2 + m^2\mathsf{M}(\xi/m)\log(\xi/m)\log(m)) && \text{if } \omega = 2 \end{aligned}$$

operations in $\mathbb{K}$.

Proof. The proof of correctness can be found in Zhou et al. (2012). We prove the cost bound following Algorithm 10 step by step.

Step **1**: since $\rho \leqslant |\mathbf{s}| \leqslant \xi$, we have $\lambda \leqslant \lceil \xi/n \rceil$.

Step **2**: using the algorithm `PM-Basis` (Giorgi et al., 2003), $\mathbf{P}$ can be computed in $\mathcal{O}(\mathsf{MM}''(m,\lambda))$ operations in $\mathbb{K}$; see Giorgi et al. (2003, Theorem 2.4). Since $\lambda \leqslant \lceil \xi/n \rceil$ and $m \in \mathcal{O}(n)$, this step uses $\mathcal{O}(\mathsf{MM}''(n,\xi/n))$ operations. Besides, from Remark 7.18 on the sum of the $\mathbf{s}$-row degrees of an $\mathbf{s}$-minimal interpolation basis, we have $|\mathrm{rdeg}_{\mathbf{s}}(\mathbf{P})| \leqslant 3n\lambda + \xi \leqslant 3(\rho + m) + \xi \leqslant 7\xi$.

Algorithm 10 *(*MinimalNullspaceBasis *(Zhou et al., 2012)).*

Input:

- matrix $\mathbf{F} \in \mathbb{K}[X]^{m\times n}$ with full rank and $m \geqslant n$,
- a shift $\mathbf{s} \in \mathbb{N}^m$ with entries in non-decreasing order and bounding the row degree of $\mathbf{F}$ componentwise.

Output:

- an $\mathbf{s}$-minimal nullspace basis $\mathbf{N}$ of $\mathbf{F}$,
- the $\mathbf{s}$-row degree of $\mathbf{N}$.

1. $\rho \leftarrow \sum_{i=m-n+1}^{m} s_i$ and $\lambda \leftarrow \lceil \rho/n \rceil$
2. $\mathbf{P} \leftarrow$ a solution to Problem 3 on input $((3\lambda,\ldots,3\lambda),\mathbf{F},\mathbf{s})$, obtained using the algorithm PM-Basis (Giorgi et al., 2003), and with the rows of $\mathbf{P}$ arranged so that $\mathrm{rdeg}_{\mathbf{s}}(\mathbf{P})$ is non-decreasing
3. Write $\mathbf{P} = [\mathbf{P}_1^{\mathsf{T}}|\mathbf{P}_2^{\mathsf{T}}]^{\mathsf{T}}$ where $\mathbf{P}_1$ consists of all rows $\mathbf{p}$ of $\mathbf{P}$ satisfying $\mathbf{pF}=0$
4. If $n=1$, Return $(\mathbf{P}_1, \mathrm{rdeg}_{\mathbf{s}}(\mathbf{P}_1))$
5. Else
 a. $\mathbf{t} \leftarrow \mathrm{rdeg}_{\mathbf{s}}(\mathbf{P}_2) - (3\lambda,\ldots,3\lambda)$
 b. $\mathbf{G} \leftarrow X^{-3\lambda}\mathbf{P}_2\mathbf{F}$
 c. Write $\mathbf{G} = [\mathbf{G}_1|\mathbf{G}_2]$ where $\mathbf{G}_1$ has $\lfloor n/2 \rfloor$ columns and $\mathbf{G}_2$ has $\lceil n/2 \rceil$ columns
 d. $(\mathbf{N}_1, \mathbf{u}) \leftarrow$ MinimalNullspaceBasis$(\mathbf{G}_1, \mathbf{t})$
 e. $(\mathbf{N}_2, \mathbf{v}) \leftarrow$ MinimalNullspaceBasis$(\mathbf{N}_1\mathbf{G}_2, \mathbf{u})$
 f. $\mathbf{N} \leftarrow [\mathbf{P}_1^{\mathsf{T}}|(\mathbf{N}_2\mathbf{N}_1\mathbf{P}_2)^{\mathsf{T}}]^{\mathsf{T}}$
 g. Return $(\mathbf{N}, (\mathrm{rdeg}_{\mathbf{s}}(\mathbf{P}_1), \mathbf{v}))$

Step **3**: finding $\mathbf{P}_1$ and $\mathbf{P}_2$ can be done by computing $\mathbf{PF}$. The matrix $\mathbf{F}$ is $m \times n$ with row degree $\mathbf{w} = \mathrm{rdeg}(\mathbf{F}) \leqslant \mathbf{s}$ (componentwise); in particular, $|\mathbf{w}| \leqslant \xi$. Besides, $\mathbf{P}$ is an $m\times m$ matrix and $|\mathrm{rdeg}_{\mathbf{w}}(\mathbf{P})| \leqslant |\mathrm{rdeg}_{\mathbf{s}}(\mathbf{P})| \leqslant 7\xi$. Then, one can augment $\mathbf{F}$ with $m-n$ zero columns and use Algorithm 3 to compute $\mathbf{PF}$; according to Proposition 4.1, this uses $\mathcal{O}(\mathrm{MM}'(m,\xi/m)) \subseteq \mathcal{O}(\mathrm{MM}'(n,\xi/n))$ operations.

Steps **5.a** and **5.b**: Computing $\mathbf{G}$ involves no arithmetic operation since the product $\mathbf{PF}$ has already been computed in Step **3**; $\mathbf{G}$ has row degree bounded by $\mathbf{t}$ (componentwise). Let us denote $\hat{m}$ the number of rows of $\mathbf{P}_2$. Because both $\mathbf{P}$ and $\mathbf{F}$ have full rank and $\mathbf{P}_1\mathbf{F} = \mathbf{0}$, $\mathbf{G}$ has full rank and at least n rows in $\mathbf{P}$ are not in the nullspace of $\mathbf{F}$, which means $n \leqslant \hat{m}$. Furthermore, according to (Zhou et al., 2012, Theorem 3.6), we have $\hat{m} \leqslant 3n/2$. Then, $\mathbf{G}$ is an $\hat{m}\times n$ matrix with $n \leqslant \hat{m} \leqslant 3n/2$ and with row degree bounded by $\mathbf{t}$. In addition, we have $\mathbf{t} \leqslant \mathbf{s}$ (Zhou et al., 2012, Lemma 3.12), and thus in particular $|\mathbf{t}| \leqslant \xi$.

Step **5.c**: for the recursive calls of Steps **5.d** and **5.e**, we will need to check that our assumptions on the dimensions, the degrees, and the rank of the input are maintained. Here, we first remark that $\mathbf{G}_1$ and $\mathbf{G}_2$ have full rank and respective dimensions $\hat{m}\times\lfloor n/2\rfloor$ and $\hat{m}\times\lceil n/2\rceil$, with $\hat{m} \geqslant \lceil n/2\rceil \geqslant \lfloor n/2 \rfloor$. Their row degrees are bounded by $\mathbf{t}$, which is in non-decreasing order and satisfies $|\mathbf{t}| \leqslant \xi$.

Step **5.d**: $\mathbf{N}_1$ is a $\mathbf{t}$-minimal nullspace basis of $\mathbf{G}_1$ and therefore it has $\hat{m} - \lfloor n/2\rfloor$ rows and $\hat{m}$ columns. Besides, $\mathbf{u} = \mathrm{rdeg}_{\mathbf{t}}(\mathbf{N}_1)$ and by (Zhou et al., 2012, Theorem 3.4), we have $|\mathbf{u}| \leqslant |\mathbf{t}| \leqslant \xi$.

Step **5.e**: we remark that $\mathbf{N}_1\mathbf{G}_2$ has $\lceil n/2\rceil$ columns and $\hat{m} - \lfloor n/2\rfloor \geqslant \lceil n/2 \rceil$ rows. We now show that it has full rank. Let us consider $\hat{\mathbf{N}}_2$ any $\mathbf{u}$-minimal nullspace basis of $\mathbf{N}_1\mathbf{G}_2$. Then $\hat{\mathbf{N}}_2$ has $\hat{m} - \lfloor n/2\rfloor - r$ rows, where r is the rank of $\mathbf{N}_1\mathbf{G}_2$. Our goal is to prove that $r = \lceil n/2\rceil$. The matrix $\hat{\mathbf{N}} = [\mathbf{P}_1^{\mathsf{T}}|(\hat{\mathbf{N}}_2\mathbf{N}_1\mathbf{P}_2)^{\mathsf{T}}]^{\mathsf{T}}$ is an $\mathbf{s}$-minimal nullspace basis of $\mathbf{F}$ (Zhou et al., 2012, Theorems 3.9 and 3.15). In particular, since $\mathbf{F}$ has full rank, $\hat{\mathbf{N}}$ has $m-n$ rows. Since $\mathbf{P}_1$ has $m-\hat{m}$ rows, this gives $m-n = m-\hat{m}+\hat{m}-\lfloor n/2\rfloor - r = m - \lfloor n/2\rfloor - r$. Thus $n = \lfloor n/2\rfloor + r$, and $r = \lceil n/2\rceil$.

Furthermore, $\mathbf{G}_2$ has row degree bounded by $\mathbf{t}$ and $\mathbf{N}_1$ has $\mathbf{t}$-row degree exactly $\mathbf{u}$, so that $\mathrm{rdeg}(\mathbf{N}_1\mathbf{G}_2) \leqslant \mathrm{rdeg}_{\mathrm{rdeg}(\mathbf{G}_2)}(\mathbf{N}_1) \leqslant \mathrm{rdeg}_{\mathbf{t}}(\mathbf{N}_1) = \mathbf{u}$. We have $|\mathbf{t}| \leqslant \xi$ and $|\mathbf{u}| \leqslant \xi$. Augmenting $\mathbf{N}_1$ and $\mathbf{G}_2$ so that they are $\hat{m}\times\hat{m}$, by Proposition 4.1, $\mathbf{N}_1\mathbf{G}_2$ can be computed using $\mathcal{O}(\mathrm{MM}'(\hat{m},\xi/\hat{m})) \subseteq \mathcal{O}(\mathrm{MM}'(n,\xi/n))$ operations. Then, $\mathbf{N}_2$ is a $\mathbf{t}$-minimal nullspace basis of $\mathbf{N}_1\mathbf{G}_2$; it has $\hat{m}-n$ rows and $\hat{m}-\lceil n/2\rceil$ columns, its $\mathbf{u}$-row degree is $\mathbf{v} = \mathrm{rdeg}_{\mathbf{u}}(\mathbf{N}_2)$, and we have $|\mathbf{v}| \leqslant |\mathbf{u}| \leqslant \xi$ (Zhou et al., 2012, Theorem 3.4).

Step **5.f**: using the previously given dimensions and degree bounds for $\mathbf{N}_1$ and $\mathbf{N}_2$, one can easily check that the product $\mathbf{N}_2\mathbf{N}_1$ can be computed by Algorithm 3 using $\mathcal{O}(\mathsf{MM}'(\hat{m}, \xi/\hat{m})) \subseteq \mathcal{O}(\mathsf{MM}'(n, \xi/n))$ operations. Now, $\mathbf{P}_2$ is $\hat{m} \times m$ with $m \geqslant \hat{m}$, and denoting $\mathbf{w}' = \mathbf{t} + (3\lambda, \ldots, 3\lambda)$, $\mathbf{P}_2$ has its row degree bounded by $\mathrm{rdeg}_{\mathbf{s}}(\mathbf{P}_2) = \mathbf{w}'$, with $|\mathbf{w}'| = |\mathrm{rdeg}_{\mathbf{s}}(\mathbf{P}_2)| \leqslant |\mathrm{rdeg}_{\mathbf{s}}(\mathbf{P})| \leqslant 7\xi$. Besides, $|\mathrm{rdeg}_{\mathbf{w}'}(\mathbf{N}_2\mathbf{N}_1)| \leqslant |\mathrm{rdeg}_{\mathbf{t}}(\mathbf{N}_2\mathbf{N}_1)| + 3(\hat{m} - n)\lambda \leqslant |\mathbf{v}| + 3n\lambda/2 \leqslant 4\xi$. Then, the product $\mathbf{N}_2\mathbf{N}_1\mathbf{P}_2$ can be computed with Algorithm 3 using $\mathcal{O}(m/\hat{m}\mathsf{MM}'(\hat{m}, \xi/\hat{m})) \subseteq \mathcal{O}(\mathsf{MM}'(n, \xi/n))$ operations, since $m \in \mathcal{O}(n)$ and $n \leqslant \hat{m}$.

Thus, we have two recursive calls with half the column dimension and the same bound ξ, and additional $\mathcal{O}(\mathsf{MM}'(n, \xi/n) + \mathsf{MM}''(n, \xi/n))$ operations for the matrix products and the computation of a minimal basis of Hermite–Padé approximants. Overall Algorithm 10 uses $\mathcal{O}(\overline{\mathsf{MM}}'(n, \xi/n) + \overline{\mathsf{MM}}''(n, \xi/n))$ operations: since $n \in \Theta(m)$, we obtain the announced cost estimate; the upper bound is a direct consequence of Lemma A.2. □

References

Alekhnovich, M., 2002. Linear Diophantine equations over polynomials and soft decoding of Reed–Solomon codes. In: FOCS'02. IEEE, pp. 439–448.

Alekhnovich, M., 2005. Linear Diophantine equations over polynomials and soft decoding of Reed–Solomon codes. IEEE Trans. Inf. Theory 51 (7), 2257–2265. http://dx.doi.org/10.1109/TIT.2005.850097.

Beckermann, B., 1990. Zur Interpolation mit polynomialen Linearkombinationen beliebiger Funktionen. PhD thesis. Department of Applied Mathematics, University of Hannover, Germany.

Beckermann, B., 1992. A reliable method for computing M-Padé approximants on arbitrary staircases. J. Comput. Appl. Math. 40 (1), 19–42. http://dx.doi.org/10.1016/0377-0427(92)90039-Z.

Beckermann, B., Labahn, G., 2000. Fraction-free computation of matrix rational interpolants and matrix gcds. SIAM J. Matrix Anal. Appl. 22 (1), 114–144. http://dx.doi.org/10.1137/S0895479897326912.

Beckermann, B., Labahn, G., 1994. A uniform approach for the fast computation of matrix-type Padé approximants. SIAM J. Matrix Anal. Appl. 15 (3), 804–823. http://dx.doi.org/10.1137/S0895479892230031.

Beckermann, B., Labahn, G., Villard, G., 2006. Normal forms for general polynomial matrices. J. Symb. Comput. 41 (6), 708–737. http://dx.doi.org/10.1016/j.jsc.2006.02.001.

Beelen, P., Brander, K., 2010. Key equations for list decoding of Reed–Solomon codes and how to solve them. J. Symb. Comput. 45 (7), 773–786. http://dx.doi.org/10.1016/j.jsc.2010.03.010.

Beelen, T.G.J., van den Hurk, G.J., Praagman, C., 1988. A new method for computing a column reduced polynomial matrix. Syst. Control Lett. 10 (4), 217–224. http://dx.doi.org/10.1016/0167-6911(88)90010-2.

Bernstein, D.J., 2011. Simplified high-speed high-distance list decoding for alternant codes. In: PQCrypto'11. In: Lect. Notes Comput. Sci., vol. 7071. Springer, pp. 200–216.

Bostan, A., Jeannerod, C.-P., Schost, E., 2008. Solving structured linear systems with large displacement rank. Theor. Comput. Sci. 407 (1–3), 155–181. http://dx.doi.org/10.1016/j.tcs.2008.05.014.

Brander, K., 2010. Interpolation and List Decoding of Algebraic Codes. PhD thesis. Technical University of Denmark.

Brent, R.P., Gustavson, F.G., Yun, D.Y.Y., 1980. Fast solution of Toeplitz systems of equations and computation of Padé approximants. J. Algorithms 1 (3), 259–295. http://dx.doi.org/10.1016/0196-6774(80)90013-9.

Busse, P., 2008. Multivariate List Decoding of Evaluation Codes with a Gröbner Basis Perspective. PhD thesis. University of Kentucky.

Cantor, D.G., Kaltofen, E., 1991. On fast multiplication of polynomials over arbitrary algebras. Acta Inform. 28 (7), 693–701. http://dx.doi.org/10.1007/BF01178683.

Chowdhury, M., Jeannerod, C.-P., Neiger, V., Schost, E., Villard, G., 2015. Faster algorithms for multivariate interpolation with multiplicities and simultaneous polynomial approximations. IEEE Trans. Inf. Theory 61 (5), 2370–2387. http://dx.doi.org/10.1109/TIT.2015.2416068.

Cohn, H., Heninger, N., 2012–2013. Approximate common divisors via lattices. In: Tenth Algorithmic Number Theory Symposium. Mathematical Sciences Publishers (MSP), pp. 271–293.

Cohn, H., Heninger, N., 2015. Ideal forms of Coppersmith's theorem and Guruswami–Sudan list decoding. Adv. Math. Commun. 9 (3), 311–339. http://dx.doi.org/10.3934/amc.2015.9.311.

Coppersmith, D., Winograd, S., 1990. Matrix multiplication via arithmetic progressions. J. Symb. Comput. 9 (3), 251–280. http://dx.doi.org/10.1016/S0747-7171(08)80013-2.

Devet, C., Goldberg, I., Heninger, N., 2012. Optimally robust private information retrieval. In: USENIX Security 12. USENIX, pp. 269–283.

Dummit, D.S., Foote, R.M., 2004. Abstract Algebra. John Wiley & Sons.

Giorgi, P., Jeannerod, C.-P., Villard, G., 2003. On the complexity of polynomial matrix computations. In: ISSAC'03. ACM, pp. 135–142.

Gupta, S., Sarkar, S., Storjohann, A., Valeriote, J., 2012. Triangular x-basis decompositions and derandomization of linear algebra algorithms over $K[x]$. J. Symb. Comput. 47 (4), 422–453. http://dx.doi.org/10.1016/j.jsc.2011.09.006.

Guruswami, V., Rudra, A., 2008. Explicit codes achieving list decoding capacity: error-correction with optimal redundancy. IEEE Trans. Inf. Theory 54 (1), 135–150. http://dx.doi.org/10.1109/TIT.2007.911222.

Guruswami, V., Sudan, M., 1999. Improved decoding of Reed–Solomon and algebraic-geometry codes. IEEE Trans. Inf. Theory 45 (6), 1757–1767. http://dx.doi.org/10.1109/18.782097.

Guruswami, V., Sudan, M., 1998. Improved decoding of Reed–Solomon and algebraic–geometric codes. In: FOCS'98, pp. 28–39.

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

Keller-Gehrig, W., 1985. Fast algorithms for the characteristic polynomial. Theor. Comput. Sci. 36, 309–317. http://dx.doi.org/10.1016/0304-3975(85)90049-0.

Knuth, D.E., 1970. The analysis of algorithms. In: Congrès Int. Math., Nice, France, vol. 3, pp. 269–274.

Kötter, R., 1996. Fast generalized minimum-distance decoding of algebraic-geometry and Reed–Solomon codes. IEEE Trans. Inf. Theory 42 (3), 721–737. http://dx.doi.org/10.1109/18.490540.

Kötter, R., Vardy, A., 2003a. Algebraic soft-decision decoding of Reed–Solomon codes. IEEE Trans. Inf. Theory 49 (11), 2809–2825. http://dx.doi.org/10.1109/TIT.2003.819332.

Kötter, R., Vardy, A., 2003b. A complexity reducing transformation in algebraic list decoding of Reed–Solomon codes. In: ITW2003. IEEE, pp. 10–13.

Le Gall, F., 2014. Powers of tensors and fast matrix multiplication. In: ISSAC'14. ACM, pp. 296–303.

Lecerf, G., 2001. Private communication.

Lee, K., O'Sullivan, M., 2006. An interpolation algorithm using Gröbner bases for soft-decision decoding of Reed–Solomon codes. In: 2006 IEEE International Symposium on Information Theory, pp. 2032–2036.

Lee, K., O'Sullivan, M.E., 2008. List decoding of Reed–Solomon codes from a Gröbner basis perspective. J. Symb. Comput. 43 (9), 645–658. http://dx.doi.org/10.1016/j.jsc.2008.01.002.

Lübbe, W., 1983. Über ein allgemeines Interpolationsproblem — lineare Identitäten zwischen benachbarten Lösungssystemen. PhD thesis. Department of Applied Mathematics, University of Hannover, Germany.

Mahler, K., 1968. Perfect systems. Compos. Math. 19 (2), 95–166.

McEliece, R.J., 2003. The Guruswami–Sudan Decoding Algorithm for Reed–Solomon Codes. IPN Progress Report 42-153.

Mulders, T., Storjohann, A., 2003. On lattice reduction for polynomial matrices. J. Symb. Comput. 35, 377–401. http://dx.doi.org/10.1016/S0747-7171(02)00139-6.

Nielsen, J.S.R., 2014. Fast Kötter–Nielsen–Høholdt interpolation in the Guruswami–Sudan algorithm. In: ACCT'14. http://arxiv.org/abs/1406.0053.

Nielsen, R.R., Høholdt, T., 2000. Decoding Reed–Solomon codes beyond half the minimum distance. In: Coding Theory, Cryptography and Related Areas. Springer, pp. 221–236.

Olshevsky, V., Shokrollahi, M.A., 1999. A displacement approach to efficient decoding of algebraic–geometric codes. In: STOC'99. ACM, pp. 235–244. http://doi.acm.org/10.1145/301250.301311.

Parvaresh, F., Vardy, A., 2005. Correcting errors beyond the Guruswami–Sudan radius in polynomial time. In: FOCS'05. IEEE, pp. 285–294.

Paszkowski, S., 1987. Recurrence relations in Padé–Hermite approximation. J. Comput. Appl. Math. 19 (1), 99–107. http://dx.doi.org/10.1016/0377-0427(87)90177-4.

Roth, R.M., Ruckenstein, G., 2000. Efficient decoding of Reed–Solomon codes beyond half the minimum distance. IEEE Trans. Inf. Theory 46 (1), 246–257. http://dx.doi.org/10.1109/18.817522.

Schönhage, A., 1971. Schnelle Berechnung von Kettenbruchentwicklungen. Acta Inform. 1, 139–144. http://dx.doi.org/10.1007/BF00289520 (in German).

Sergeyev, A.V., 1987. A recursive algorithm for Padé–Hermite approximations. USSR Comput. Math. Math. Phys. 26 (2), 17–22. http://dx.doi.org/10.1016/0041-5553(86)90003-0.

Storjohann, A., 2000. Algorithms for Matrix Canonical Forms. PhD thesis. Swiss Federal Institute of Technology – ETH.

Storjohann, A., 2006. Notes on computing minimal approximant bases. In: Challenges in Symbolic Computation Software. Dagstuhl Seminar Proceedings. http://drops.dagstuhl.de/opus/volltexte/2006/776.

Sudan, M., 1997. Decoding of Reed–Solomon codes beyond the error-correction bound. J. Complex. 13 (1), 180–193. http://dx.doi.org/10.1006/jcom.1997.0439.

Van Barel, M., Bultheel, A., 1991. The computation of non-perfect Padé–Hermite approximants. Numer. Algorithms 1 (3), 285–304. http://dx.doi.org/10.1007/BF02142327.

Van Barel, M., Bultheel, A., 1992. A general module theoretic framework for vector M-Padé and matrix rational interpolation. Numer. Algorithms 3, 451–462. http://dx.doi.org/10.1007/BF02141952.

von zur Gathen, J., Gerhard, J., 2013. Modern Computer Algebra, third edition. Cambridge University Press.

Welch, L.R., Berlekamp, E.R., Dec. 30 1986. Error correction for algebraic block codes. US Patent 4,633,470.

Zeh, A., 2013. Algebraic Soft- and Hard-Decision Decoding of Generalized Reed–Solomon and Cyclic Codes. PhD thesis. École Polytechnique.

Zeh, A., Gentner, C., Augot, D., 2011. An interpolation procedure for list decoding Reed–Solomon codes based on generalized key equations. IEEE Trans. Inf. Theory 57 (9), 5946–5959. http://dx.doi.org/10.1109/TIT.2011.2162160.

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

Zhou, W., Labahn, G., 2012. Efficient algorithms for order basis computation. J. Symb. Comput. 47 (7), 793–819. http://dx.doi.org/10.1016/j.jsc.2011.12.009.

Zhou, W., Labahn, G., Storjohann, A., 2012. Computing minimal nullspace bases. In: ISSAC'12. ACM, pp. 366–373.