Fraction-free Computation of Matrix Rational Interpolants and Matrix GCDs

Bernhard Beckermann
Laboratoire d'Analyse Numérique et d'Optimisation,
Université des Sciences et Technologies de Lille,
59655 Villeneuve d'Ascq Cedex, France
e–mail: bbecker@ano.univ-lille1.fr
and
George Labahn
Department of Computer Science
University of Waterloo, Waterloo, Ontario, Canada
e–mail: glabahn@daisy.uwaterloo.ca

Abstract

We present a new set of algorithms for computation of matrix rational interpolants and one-sided matrix greatest common divisors. Examples of these interpolants include Padé approximants, Newton-Padé, Padé-Hermite, simultaneous Padé approximants and more generally M-Padé approximants along with their matrix generalizations. The algorithms are fast and compute all solutions to a given problem. Solutions for all (possibly singular) subproblems along offdiagonal paths in a solution table are also computed by stepping around singular blocks on some path corresponding to "closest" regular interpolation problems.

The algorithms are suitable for computation in exact arithmetic domains where growth of coefficients in intermediate computations are a central concern. This coefficient growth is avoided by using fraction-free methods. At the same time the methods are fast in the sense that they are at least an order of magnitude faster than existing fraction-free methods for the corresponding problems. The methods make use of linear systems having a special striped Krylov structure.

Key words: Hermite Padé approximant, simultaneous Padé approximant, striped Krylov matrices, Fraction-free arithmetic

Subject Classifications: AMS(MOS): 65D05, 41A21, CR: G.1.2

1 Introduction

A number of methods are available for the computation of various rational interpolation problems. Consider, for example, the simplest case of rational interpolation, that of Padé approximation. One can compute a Padé approximant by setting up a linear system of equations and use Gaussian elimination to solve the system. The number of operations in this case is $O(n^3)$ where n is the number of equations in the system. However, since the coefficient matrix of this system has a special Hankel or Toeplitz structure, there exist more efficient algorithms for these computations. Examples include fast $O(n^2)$ algorithms and even superfast $O(n\ log^2 n)$ algorithms (cf., Brent, Gustavson and Yun [14] or Cabay and Choi [17] in addition to many others). A similar statement can also be made for other matrix-like Padé approximation problems. Here one finds fast or superfast algorithms for computing Hermite-Padé and simultaneous Padé approximants, e.g., Van Barel and Bultheel [51, 52], Cabay, Labahn and Beckermann [18], Cabay and Labahn [21], and Beckermann and Labahn [7, 8, 9]. In all the examples above the algorithms are both fast and avoid problems associated to the existence of singular blocks in an associated solution table. Alternatively, one may obtain fast algorithms for Padé approximation by translating to polynomial language some of the algorithms developed for structured matrices having a small displacement rank, for example those found in Heinig and Rost [35].

However, at the implementation level, these algorithms have drawbacks that limit their effectiveness. For example suppose one is working in a floating point environment. Since the previously mentioned algorithms assume exact arithmetic, implementations in floating point domains do not take into consideration roundoff error. In these cases the computations all suffer from some degree of numerical instability. It is only recently that a number of new algorithms have appeared that are both fast and stable in a numerical setting, for example [6, 19, 22, 33, 53] for Padé problems, [12, 13, 34, 26, 27] for Toeplitz and Hankel systems, [23, 30, 32] and further references mentioned in [31] for systems with displacement structure.

The round-off problems encountered when implementing in floating point domains do not appear when implementing in exact arithmetic environments, for example in computer algebra systems such as Maple or Mathematica. However, even in these cases it turns out that most existing algorithms have problems that also limit their usefulness. In the case of numerical arithmetic, the efficient algebraic algorithms are fast but sometimes suffer from a lack of accuracy. In exact domains these algorithms are accurate but often lack efficiency. For example, in Czapor and Geddes [25], it is shown that a minor modification of Gaussian elimination in fact computes Padé approximants more efficiently than Levinson's algorithm, that is, in this case an $O(n^3)$ algorithm is faster than an $O(n^2)$ algorithm. The reason for this is simple to explain: in exact arithmetic domains, operations such as addition or multiplication do not have a constant cost. Rather the arithmetic cost depends on the size of the components and so we need to measure bit complexity rather

than operations complexity. The (possibly exponential) growth in the cost of intermediate arithmetic operations may be observed in particular when the domain of coefficients is a field of quotients $\mathbf{Q}(a_1, \cdots, a_n)$ where $\mathbf{Q}$ is the field of rational numbers (or an algebraic extension of the rational numbers) and $a_1, \cdots, a_n$ are indeterminants, a typical situation for symbolic computation in computer algebra systems. In order to compute in these domains one must try for a low complexity while at the same time keeping the components of the arithmetic operations at a small size. In addition, the cost of keeping the components of the arithmetic operations at a small size must be done in an efficient manner.

In this paper we present a new fast algorithm for efficiently computing *all* solutions to a variety of matrix rational interpolation problems along with one-sided matrix greatest common divisors. The interpolation problems covered include the partial realization problem for matrix power series and Padé, Newton-Padé, Padé-Hermite, simultaneous Padé, M-Padé and Multi-point Padé approximation problems and their matrix generalizations. The connection between rational interpolation and greatest common divisor problems has been known for a long time and has been successfully exploited in the scalar case.

The algorithm is recursive, providing also solutions for all (possibly singular) subproblems along offdiagonal paths in a solution table. Here singular subproblems are not skipped over via pseudodivisions or look–ahead techniques, but following [5, 8, 51] we step around singular blocks on some path corresponding to "closest" regular interpolation problems. This leads to an additional gain in complexity if there are only few regular subproblems, a rather typical situation for GCD computations.

Rather than present the algorithm for a field, we assume that the coefficient domain is an integral domain and give a *fraction-free* algorithm for efficiently computing solutions to these interpolation problems. The concept of fraction-free implies that arithmetic operations remain inside the integral domain, rather than requiring that one does arithmetic in its quotient field. This avoids the need for costly greatest common divisor computations required for such rational operations making the algorithm suitable for implementation in computer algebra systems. This allows for efficient computation of matrix interpolation problems in the case of parameterized data. Such computations also appear in such diverse applications as the Gfun package of Salvy and Zimmerman [49] for determining recurrences relations, factorization of linear differential operators [54] and computation of matrix normal forms [55].

The algorithm presented here is at least an order of magnitude faster than applying the fraction-free algorithm of Bareiss [2] which is based on Gaussian elimination. This is the only known fraction-free method that will also work for the rational interpolation problems studied here. However there have been fraction-free algorithms that are faster than Bareiss's algorithm in some special cases. For Padé approximation the algorithm of Cabay and Kossowski [20] makes use of the close relationship between Padé approxima-

tion and polynomial remainder sequences to obtain an improved fraction-free algorithm. For matrix Padé approximation the algorithm of Beckermann, Cabay and Labahn [10] uses a recursive procedure based on modified Schur complements of the associated linear equations to improve on Gaussian elimination. Finally the subresultant GCD algorithm of Brown and Collins [15, 24] gives a fast greatest common divisor algorithm in the case of scalar polynomials. In all cases our algorithm is also faster or at least as fast as those mentioned in special cases.

In terms of linear algebra, we can view our problem as determining nullspaces of rectangular striped Krylov matrices and their principal submatrices in a fast and fraction-free manner. Notice that this task includes the fraction-free computation of vectors required for explicit inversion formulas, for example for Hankel, Toeplitz or Sylvester matrices and their block counterparts [35] (for an interpretation in terms of Padé problems see for instance [39, 42]). In the regular case where all principal submatrices are nonsingular, it is possible to look for fraction–free counterparts of known algorithms for structured matrices, for example fast Gaussian elimination scheme, that is, Schur type algorithms, or Levinson type methods [31, 35]. Recent extensions [23, 30, 31, 32] also allow for pivoting in order to overcome problems with singularities. However, in our setting, additional transformation techniques would be necessary in order to allow for pivoting. Our alternate approach is motivated by the facts that transformations may lead to a significant increase of complexity of the input data and in any case cannot be done in a fraction-free manner. In addition, the kind of pivoting used in these extensions does not allow us to solve subproblems of our initial interpolation problem.

The rest of the paper is organized as follows. Section 2 introduces the rational interpolation problems defined in terms of a "special rule". Section 3 shows that the rational interpolation problems can be interpreted in linear algebra terms as solving a linear systems of equations having a striped Krylov matrix as a coefficient matrix. Some regularity properties are studied in Section 4 while Section 5 introduces Mahler systems, a matrix of determinant polynomials which give a basis for our solution spaces. Section 6 gives a fraction-free recursion along a so called perfect path while Section 7 considers the more difficult non-perfect case. Section 8 shows how the algorithm from the previous section can be used to compute the one-sided GCD of two matrix polynomials. The last section includes a conclusion along with a discussion of future research directions.

2 Rational Interpolation and their Linear Systems

Let $\mathbb{D}$ be an integral domain with $\mathbb{Q}$ its quotient field. Let $\mathcal{V}$ be an infinite dimensional vector space over $\mathbb{Q}$ having a basis $(\omega_u)_{u=0,1,\ldots}$ with $(c_u)_{u=0,1,\ldots}$ its dual basis (i.e. a set of linear functionals on $\mathcal{V}$ satisfying $c_u(\omega_v) = \delta_{u,v}$). Thus every element f of $\mathcal{V}$ can be

written as

$$f = f_0 \cdot \omega_0 + f_1 \cdot \omega_1 + f_2 \cdot \omega_2 + \cdots \tag{1}$$

with $c_u(f) = f_u$. We define the *Order* of a nontrivial element f of $\mathcal{V}$ by

$$ord(f) = n \quad \text{iff} \quad c_0(f) = \cdots = c_{n-1}(f) = 0 \text{ and } c_n(f) \neq 0,$$

and $ord(0) = +\infty$.

We also assume that we have a *special* element z that acts on $\mathcal{V}$ via a special multiplication rule

$$c_u(z \cdot f) = c_{u,0} \cdot c_0(f) + \cdots + c_{u,u} \cdot c_u(f) \text{ with } c_{u,v} \in \mathbb{D}\,. \tag{2}$$

This special rule can be viewed as a type of Leibniz chain rule. The special rule allows us to define a multiplication $p(z) \cdot f$ for any polynomial $p \in \mathbb{Q}[z]$ and $f \in \mathcal{V}$ making $\mathcal{V}$ an infinite dimensional module over $\mathbb{Q}[z]$.

In this paper we will study the following interpolation problem with polynomial linear combinations of functions $f^{(1)}, \ldots, f^{(m)}$ where $m \geq 2$.

Definition 2.1 (Rational Interpolation Problem)
Let $f = [f^{(1)}, \ldots, f^{(m)}]$ be a vector of m elements from $\mathcal{V}$, σ a positive integer and $\vec{n} = (\vec{n}^{(1)}, \ldots, \vec{n}^{(m)})$ a multi-index. Determine a vector $p(z) = [p^{(1)}(z), \ldots, p^{(m)}(z)]^T$ of polynomials in z, with each $p^{(\ell)}(z)$ having degree bounded by $\vec{n}^{(\ell)} - 1$, and satisfying the order condition

$$\text{ord}\,(f \cdot p(z)) = \text{ord}\,(f^{(1)} \cdot p^{(1)}(z) + \cdots + f^{(m)} \cdot p^{(m)}(z)) \geq \sigma. \tag{3}$$

In this case, $p(z)$ will be referred to as solution of type $(\sigma, \vec{n})$. □

Example 2.2 (Hermite–Padé approximants [45, 46, 47, 48, 51])
Let $\mathcal{V}$ be the space $\mathbb{Q}[[z]]$ of formal power series around 0 with basis $(z^u)_{u=0,1,\ldots}$ and let the $c_{i,j}$ be defined by $c_{i,j} = \delta_{i-1,j}$. Then the special multiplication rule is simply the standard multiplication by z. With $\sigma = |\vec{n}| - 1$ where $|\vec{n}| := \vec{n}^{(1)} + \cdots + \vec{n}^{(m)}$, the interpolation problem (3) is the Hermite-Padé approximation problem of type $\vec{n}$, introduced by Hermite in 1873. When $m = 2$ and $f^{(2)} = -1$, this gives the classical Padé approximant. Hermite–Padé approximation also includes other classical approximation problems such as algebraic approximants ($f = (1, g, g^2, \ldots, g^{m-1})$) and G^3J approximants ($m = 3, f = (g', g, 1)$). We refer the reader to [1] for additional examples. □

Before giving further examples for the rational interpolation problem of Definition 2.1, let us have a closer look at the underlying system of linear equations. Notice first that we may rewrite the special multiplication rule (2) in terms of linear algebra. We denote by $\mathbf{C} = (c_{u,v})_{u,v=0,1,\ldots}$ the lower triangular infinite matrix determined by the coefficients of (2), and by $\mathbf{C}_\sigma$, $\sigma \geq 0$ its principal submatrix of order σ. Furthermore, for each $f \in \mathcal{V}$ and nonnegative integer σ we associate a vector of coefficients

$$\mathbf{F}_\sigma = [c_0(f), \ldots, c_{\sigma-1}(f)]^T, \quad \mathbf{F} = [c_0(f), c_1(f), c_2(f), \ldots]^T. \tag{4}$$

Note that we begin our row and column enumeration at 0. Then in matrix terms the special multiplication rule can be interpreted as

$$\mathbf{C}_\sigma \cdot \mathbf{F}_\sigma = [c_0(z \cdot f), \ldots, c_{\sigma-1}(z \cdot f)]^T$$

and more generally

$$p(\mathbf{C}_\sigma) \cdot \mathbf{F}_\sigma = [c_0(p(z) \cdot f), \ldots, c_{\sigma-1}(p(z) \cdot f)]^T$$

for any polynomial $p(z) \in \mathbb{Q}[z]$ and for any nonnegative integer σ.

For our rational interpolation problem we can associate as in (4) to f the vectors of values $\mathbf{F}_\sigma = (\mathbf{F}_\sigma^{(1)}, ..., \mathbf{F}_\sigma^{(m)})$, $\mathbf{F}_\sigma^{(i)} = [c_0(f^{(i)}), \ldots, c_{\sigma-1}(f^{(i)})]^T$, $i = 1, .., m$. Then the order condition (3) in Definition 2.1 may be rewritten as

$$p^{(1)}(\mathbf{C}_\sigma) \cdot \mathbf{F}_\sigma^{(1)} + \ldots + p^{(m)}(\mathbf{C}_\sigma) \cdot \mathbf{F}_\sigma^{(m)} = 0.$$

In order to obtain explicitly a system of equations, we introduce

$$\mathbf{K}(\vec{n}, \mathbf{C}_\sigma, \mathbf{F}_\sigma) = \left[\begin{array}{cccc|c|cccc} \mathbf{F}_\sigma^{(1)} & \mathbf{C}_\sigma \mathbf{F}_\sigma^{(1)} & \cdots & \mathbf{C}_\sigma^{\vec{n}^{(1)}-1} \mathbf{F}_\sigma^{(1)} & \cdots & \mathbf{F}_\sigma^{(m)} & \cdots & \mathbf{C}_\sigma^{\vec{n}^{(m)}-1} \mathbf{F}_\sigma^{(m)} \end{array} \right],$$

a striped Krylov matrix of size $\sigma \times |\vec{n}|$. Furthermore, we identify a vector polynomial $p(z) = [p^{(1)}(z), \ldots, p^{(m)}(z)]^T$ of the form $p^{(i)}(z) = \sum_{j=0}^{\vec{n}^{(i)}-1} p_j^{(i)} z^j$, $i = 1, .., m$ with its *coefficient vector*

$$\mathbf{P} = [p_0^{(1)}, ..., p_{\vec{n}^{(1)}-1}^{(1)} | \ldots\ldots | p_0^{(m)}, ..., p_{\vec{n}^{(m)}-1}^{(m)}]^T.$$

Then $p(z)$ is a solution of type $(\sigma, \vec{n})$ if and only if its coefficient vector $\mathbf{P}$ satisfies

$$\mathbf{K}(\vec{n}, \mathbf{C}_\sigma, \mathbf{F}_\sigma) \cdot \mathbf{P} = 0. \tag{5}$$

In the remaining part of this section, further special cases of the interpolation problem of Definition 2.1 are discussed.

Example 2.3 (Vector and Power Hermite–Padé approximants [7, 8, 52])
Let $\mathcal{V}$ be the space $\mathbb{Q}^s[[z]]$ of $1 \times s$ vectors of formal power series around 0. A basis

for $\mathcal{V}$ is given by $\omega_u = \omega_{n\cdot s+k} = z^n \cdot \vec{e}_{k+1}$ with $0 \le k < s$, where $\vec{e}_k$ denotes the k-th unit vector. Let the $c_{i,j}$ be defined by $c_{i,j} = \delta_{i-s,j}$. Then the special multiplication rule is again the standard scalar multiplication by z, viewed as a scalar. In this case, problem (3) with $\sigma = |\vec{n}| - 1$ is the vector Hermite-Padé approximation problem of type $\vec{n}$. This interpolation problem appears for example, in the new Van Hoeij algorithm for the factorization of differential operators [54].

We can also let $\mathcal{V}$ be the space $\mathbb{Q}[[x]]$ of formal power series around 0 with basis $\tilde{\omega}_u = \omega_u(x^s) \cdot [1, x, ..., x^{s-1}]^T$ with the ω_u from above. Let the $c_{i,j}$ again be defined by $c_{i,j} = \delta_{i-s,j}$. Then the special rule is multiplication by $z = x^s$. In this case, problem (3) is then to find polynomials $p^{(i)}$ in z with the correct degree bounds (with respect to z of course) and satisfying the equation

$$f^{(1)} \cdot p^{(1)}(x^s) + \cdots + f^{(m)} \cdot p^{(m)}(x^s) = r_\sigma x^\sigma + r_{\sigma+1} x^{\sigma+1} + \cdots .$$

This is the Power Hermite-Padé approximation problem. Note that this problem is the same a the first part of our example obtained by multiplying both sides of every basis equation (1) by the vector $[1, x, ..., x^{s-1}]^T$. This is the "s-trick" described in [7, 8]. Besides vector Hermite-Padé approximants, Power Hermite-Padé approximation can be used to represent (and hence to compute) matrix Padé approximants [41] and simultaneous Padé approximants [45] along with their matrix generalizations [39]. For instance, solutions of type $(|\vec{n}| - s, \vec{n})$ are required as building block for Matrix Padé approximants (see [8]). □

Example 2.4 (Linearized rational interpolation)
Suppose that we have a sequence of not necessarily distinct knots $x_i \in \mathbb{D}$, and a function g being sufficiently smooth in a neigborhood of these knots. The linearized rational interpolation problem of type $[L/M]$ (see, e.g., [1]) consists in finding polynomials p and q of degree at most L, and M, respectively, such that $-p + g \cdot q = [-1, g] \cdot [p, q]^T$ vanishes at $x_0, ..., x_{L+M}$, counting multiplicities.

Let $\mathcal{V}$ be the space of all formal Newton series in z with respect to the given knots $x_0, x_1, ...$ Note that a basis of $\mathcal{V}$ (or some finite dimensional counterpart) may be constructed using either Newton, Lagrange, or Hermite polynomials. Therefore, there are several choices for the sequence of linear functionals $(c_u)_{u=0,1,2,...}$ in order to reformulate the linearized rational interpolation problem using the formalism of Definition 2.1. For instance, one may take as c_v the v-th divided difference $[x_0, \ldots, x_v]$. It is easy to verify that for these linear functionals the special multiplication rule (2) holds, with $c_{i,j} = \delta_{i,j} \cdot x_i + \delta_{i-1,j}$, $i > 0$, and $c_{0,0} = x_0$.

If the knots $x_0, x_1, ..$ are distinct, then the simpler choice $c_v(g) = g(x_v)$ leads to the special multiplication rule (2) with $c_{i,j} = \delta_{i,j} \cdot x_i$. In the case of not necessarily

distinct knots, we may more generally consider the values of the successive derivatives, i.e., $c_v(g) = g^{(\rho_v)}(x_v)/(\rho_v!)$, where ρ_v denotes the multiplicity of x_v in $(x_0, x_1, ..., x_{v-1})$. Here the components $c_{i,j}$ for the special multiplication rule is based on (some permutation of) a Jordan normal form matrix $\mathbf{C}$. □

In Example 2.4 we mentioned the case $m = 2$ with $f = [-1, g]$. The case of general f has also been discussed by several authors.

Example 2.5 (M-Padé approximants [3, 4, 5, 44, 45])
Suppose that we have a sequence of not necessarily distinct knots $x_i \in \mathbb{D}$. Let again $\mathcal{V}$ be the space of all formal Newton series in z with bases elements $\omega_u = (z - x_0) \cdots (z - x_{u-1})$, with the dual basis consisting of the v-th divided difference $c_v = [x_0, \ldots, x_v]$, $v \geq 0$ (the corresponding special multiplication rule is given in Example 2.4). Solutions of type $(|\vec{n}| - 1, \vec{n})$ of our interpolation problem of Definition 2.1 are known as M-Padé approximants of type $\vec{n}$. One can also obtain a vector M-Padé problem using the same method as described in Example 2.3.

An important application for M–Padé approximation is the generalized Richardson extrapolation process (GREP) where one tries to approximate the limit of some sequence $(g(x_j))_{j=0,1,...}$ with distinct $x_0, x_1, ...$ by interpolating with help of the function 1 and polynomial linear combinations of some functions $g_1, ..., g_m$ [50]. Here the sequence of knots and the functions $g_1, ..., g_m$ are chosen such that $(x_j^\ell \cdot g_i(x_j))_{j=0,1,...}$ tends to zero for all i, ℓ. Thus the (scalar) ratio between the first and the second component of an M–Padé approximant of type $[1, 1, n_1, .., n_m]$ with respect to the system $[-1, g, g_1, .., g_m]$ is used for approximating the desired limit. Note that, due to the available data, the linear functionals $c_v(f) = f(x_v)$ may be preferable. □

Example 2.6 (Controller–form realizations)
In some applications like controller–form realizations [36, Section 6.4.2], one aims for a representation of $(z\mathbf{I} - \mathbf{C})^{-1}\mathbf{F}$. Since $\mathbf{C}$ is lower triangular, its first σ components are given by $(z\mathbf{I}_\sigma - \mathbf{C}_\sigma)^{-1} \cdot \mathbf{F}_\sigma$. We look for a representation as an $\sigma \times m$ valued rational function of the form $\Psi(z) \cdot D(z)^{-1}$, with $\Psi(z), D(z)$ being matrix polynomials, and $D(z) = D_0 + D_1 z^1 + ... + D_N z^N$ being nonsingular, that is, the polynomial $det\, D(z)$ does not vanish identically. Taking as $\Psi(z)$ the polynomial part of $R(z) := (z\mathbf{I}_\sigma - \mathbf{C}_\sigma)^{-1} \cdot \mathbf{F}_\sigma \cdot D(z)$, it remains to determine the integer N and the coefficients of $D(z)$ such that the coefficients of the negative powers of the Laurent expansion around infinity of $R(z)$ vanish, or, equivalently,

$$\mathbf{F}_\sigma \cdot D_0 + \mathbf{C}_\sigma \cdot \mathbf{F}_\sigma \cdot D_1 + ... + \mathbf{C}_\sigma^N \cdot \mathbf{F}_\sigma \cdot D_N = 0.$$

This means that we are looking for m solutions of type $(\sigma, \vec{n})$, $\vec{n} = [N + 1, ..., N + 1]$, being linearly independent over $\mathbb{Q}[z]$. In addition, one generally wishes to minimize the

degree of $det\, D(z)$ in order to have so–called *minimal realizations*. Thus we are left with the problem of computing bases of the set of all solutions of order $\geq \sigma$, following ideas exploited already in previous papers (compare, e.g., [9, Lemma 2.5]). Finally notice that these requirements do not fix a unique solution $D(z)$. In fact one may impose some additional structure such as $D(z)$ being in Hermite or Popov normal form [36, Section 6.7]. A generalization of these normal forms will be studied in Section 7. □

3 The linear algebra background

For the remainder of this paper we will assume that we have a fixed lower triangular infinite matrix $\mathbf{C}$ and a fixed $\mathbf{F} = [\mathbf{F}^{(1)}, \cdots, \mathbf{F}^{(m)}]$ of infinite coefficient vectors for elements $f^{(1)}, \cdots, f^{(m)}$ of $\mathcal{V}$. Let $\vec{n}$ be a multi-index and σ a positive integer. In order to simplify notation, we will simply drop $\mathbf{C}_\sigma$ and $\mathbf{F}_\sigma$ from our notation when using the striped Krylov matrices, i.e., we will write $\mathbf{K}(\vec{n}, \sigma) = \mathbf{K}(\vec{n}, \mathbf{C}_\sigma, \mathbf{F}_\sigma)$ for the associated striped Krylov matrix of size $\sigma \times |\vec{n}|$. Note that, since $\mathbf{C}$ is lower triangular, the matrix $\mathbf{K}(\vec{n}, j)$ for $j < \sigma$ consists of the first j rows of $\mathbf{K}(\vec{n}, \sigma)$.

We have seen in Section 2 that finding a solution p of type $(|\vec{n}| - 1, \vec{n})$ of the interpolation problem of Definition 2.1 with exact order $|\vec{n}| - 1$ is equivalent to solving the system of linear equations

$$\mathbf{K}(\vec{n}, |\vec{n}|) \cdot \bar{\mathbf{P}} = [0, \ldots, 0, 1]^T \tag{6}$$

for the corresponding coefficient vector $\bar{\mathbf{P}}$. In our case, we look for solutions with coefficients not in the fraction field $\mathbb{Q}$ but in the integral domain $\mathbb{D}$. This is accomplished by means of Cramer's rule over $\mathbb{Q}$, giving a solution

$$\mathbf{K}(\vec{n}, |\vec{n}|) \cdot \mathbf{P} = det\,(\mathbf{K}(\vec{n}, |\vec{n}|)) \cdot [0, \ldots, 0, 1]^T \tag{7}$$

with $\mathbf{P}$ being a vector having only coefficients from $\mathbb{D}$. Here, the determinant representation of $\mathbf{P}$ furnished by Cramer's rule is quite useful and will be studied in Section 5. For instance, this representation enables us to obtain bounds for the size (in bits) of such a solution in terms of the initial size of the components of the series using Hadamard's inequality [29, p.299]

$$|det\,(a_{j,k})| \leq \prod_j [\sum_k |a_{j,k}|^2]^{1/2}. \tag{8}$$

In fact, Cramer solutions are also furnished by applying fraction-free Gaussian elimination [2, 29] on (6). Our contribution is to show in the second part of this paper that Cramer solutions may be obtained in a more efficient way.

It seems that in general Cramer solutions may be considered as the "simplest" solutions of (6) with coefficients in $\mathbb{D}$. Of course, one may construct examples where

additional simplifications occur, but it can be quite expensive to detect such further simplifications. To illustrate this statement, take for instance the problem of computing a scalar GCD. Here several methods exist which avoid fractions (for a summary see, e.g., [29, Section 7.2]), for instance the reduced PRS. However, only the subresultant GCD algorithm of Brown and Collins [15, 24] gives "maximal" Cramer solutions.

We recall that, depending on the matrix $\mathbf{C}$ defined by our special rule (2), we may obtain a system of equations with a matrix of coefficients having a quite particular structure, for instance the following.

Example 3.1 (Toeplitz and generalized Sylvester matrices)
Let $\mathbf{C}$ be the classical lower shift matrix, that is, $c_{i,j} = \delta_{i-1,j}$. Then $\mathbf{K}(\vec{n}, \sigma)$ is a generalized Sylvester matrix [39] with each stripe a lower triangular Toeplitz matrix. If $m = 2$ and

$$\mathbf{F} = \begin{bmatrix} p_0 & \cdots & p_k & 0 & \cdots & 0 \\ q_0 & \cdots & \cdots & q_n & 0 \cdots & 0 \end{bmatrix}^T$$

then

$$\mathbf{K}((n,k), n+k) = \left[\begin{array}{cccc|cccc} p_0 & 0 & \cdots & 0 & q_0 & 0 & \cdots & 0 \\ & p_0 & \ddots & \vdots & & q_0 & \ddots & \vdots \\ \vdots & & \ddots & 0 & \vdots & & \ddots & 0 \\ \vdots & & & p_0 & \vdots & & & q_0 \\ p_k & & & \vdots & q_n & & & \vdots \\ 0 & \ddots & & \vdots & 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & p_k & 0 & \cdots & 0 & q_n \end{array}\right]$$

is the classical Sylvester matrix for the polynomials $p(z) = \sum_{i=0}^{k} p_{k-i} z^i$ and $q(z) = \sum_{i=0}^{n} q_{n-i} z^i$. Sylvester's matrix is heavily used in the fraction–free computation of the GCD of two polynomials (cf. [29]). □

Beside (striped) Toeplitz or Sylvester matrices associated to (Hermite–)Padé approximation, striped Krylov matrices with lower triangular $\mathbf{C}$ may be used to represent other well–known structured matrices. For instance, for vector or power Hermite–Padé approximants (Example 2.3) we may choose as $\mathbf{C}$ the s-th power of the lower shift matrix. Then $\mathbf{K}(\vec{n}, \sigma)$ is a generalized vector Sylvester matrix with each stripe a vector Toeplitz matrix having $s \times 1$ vector entries. If all the stripes have equal length k then this is, up to permutation, the same as a block triangular Toeplitz matrix with blocks of size $s \times k$. We can also consider the case where $\mathbf{C}$ is a matrix made up of diagonal blocks of (possibly different sized) shift matrices, leading to mosaic generalized Sylvester matrices.

In case of the rational interpolation problems discussed in Examples 2.4 and 2.5, one is left with matrices $\mathbf{C}$ consisting of diagonal blocks of the form

$$\begin{bmatrix} x_0 & 0 & \cdots & 0 \\ 0 & x_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & x_k \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} x_0 & 0 & \cdots & \cdots & 0 \\ 1 & x_1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & & \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & x_k \end{bmatrix},$$

the first in the case of function evaluations, the second if one uses divided differences (or simply successive derivatives at a point different from zero). For the first choice, $\mathbf{K}(\vec{n}, \sigma)$ consists of stripes, each of them a rectangular Vandermonde matrix multiplied on the left by a diagonal matrix.

A powerful formalism for solving structured systems is the concept of displacement operators (see, for example [35]), that is, matrices $\mathbf{M}$ where, for some given matrices $\mathbf{A}_1, \mathbf{A}_2$, the quantity $\mathbf{A}_1\mathbf{M} - \mathbf{M}\mathbf{A}_2$ has a much smaller rank than the size of $\mathbf{M}$. In our case we have

$$\begin{aligned} &\mathbf{C}_\sigma \cdot \mathbf{K}(\vec{n}, \sigma) - \mathbf{K}(\vec{n}, \sigma) \cdot \mathbf{Z} \\ &= \left[\begin{array}{ccccc|c|ccccc} 0 & \cdots & 0 & \mathbf{C}_\sigma^{\vec{n}^{(1)}} \mathbf{F}_\sigma^{(1)} & & \cdots & & 0 & \cdots & 0 & \mathbf{C}_\sigma^{\vec{n}^{(m)}} \mathbf{F}_\sigma^{(m)} \end{array} \right], \end{aligned}$$

where $\mathbf{Z}$ is block diagonal consisting of lower shift matrices of size $\vec{n}^{(j)}, j = 1, 2, .., m$. Thus our striped Krylov matrices $\mathbf{K}(\vec{n}, \sigma)$ have displacement rank m.

An important number of fast (but not fraction-free) algorithms have been suggested in the last years for factoring or inverting matrices with small displacement rank, or for solving corresponding structured systems. For instance we mention Levinson type methods based on bordering techniques and Schur type algorithms (also called fast Gaussian elimination) based on the fact that Schur complements verify similar displacement equations [31, 35]. In our case, we wish to have a (fraction–free) description of the nullspace of all principal submatrices of $\mathbf{K}' := \mathbf{K}(\vec{n}, \sigma)\mathbf{P}$, where $\mathbf{P}$ is some permutation matrix such that $\mathbf{Z}' := \mathbf{P}^T\mathbf{Z}\mathbf{P}$ remains strictly lower triangular (that is, we follow some path in the corresponding solution table). Notice that the displacement equation for Schur complements (cf. [31, Lemma 3.1]) becomes quite involved since $\mathbf{Z}'$ is no longer upper triangular. Also, in case of singularities, one has to use look–ahead, or one needs to add a technique of pivoting [23, 30, 31, 32] which for general displacement operators $\mathbf{A}_1\mathbf{M} - \mathbf{M}\mathbf{A}_2$ seems to be only feasible if one of the matrices $\mathbf{A}_1$ (for row pivoting) or $\mathbf{A}_2$ (for column pivoting) is diagonal. However, our matrix $\mathbf{C}$ will be diagonal only in special cases, [1] and $\mathbf{Z}'$ is never diagonal. One usually overcomes this drawback by using transformation techniques, that is, by multiplying $\mathbf{K}'$ on the left and/or on the right by suitable matrices (e.g., FFT matrices in the Toeplitz case) which changes the displacement operator but keeps the displacement rank essentially invariant [30, 31].

[1] See Examples 2.4 and 2.5. Here a row pivoting corresponds to a permutation of interpolation points (which need to be distinct), a classical technique in rational interpolation or M-Padé approximation.

In the present paper we use neither transformation techniques nor look–ahead. In both cases these methods may present major inconveniences. Transformations can lead to a significant increase of complexity of the input data, and look–ahead is less efficient for large jumps (a common occurrence in GCD problems). In addition, both approaches do not allow us to keep track of all interpolation subproblems corresponding to principal submatrices of $\mathbf{K}'$. Our contribution in Section 7 is to show that a very particular column pivoting still enables us to solve all corresponding subproblems. Here we generalize polynomial recurrences presented by several authors [8, 45, 46, 47, 51], and thus a polynomial language is more appropriate in our context.

4 Normality and controllable data

For the solvability of system (6) we require some regularity assumptions. The aim of this section is to discuss several such concepts.

Definition 4.1 (Multigradients, Normality)
The scalar

$$d(\vec{n}) = det(\mathbf{K}(\vec{n}, |\vec{n}|))$$

is called the multigradient of $\mathbf{F}$ *of type* $\vec{n}$. *The multi-index* $\vec{n}$ *is called a* normal *point if* $d(\vec{n}) \neq 0$. *Finally, the data* $(\mathbf{C}, \mathbf{F})$ *is called* perfect *if every multi-index is normal.* □

We use the convention that the determinant of an empty matrix equals one. Of course, given $\sigma > 0$, the existence of a normal point $\vec{n}$ with $|\vec{n}| = \sigma$ requires that the linear functionals $c_0, .., c_{\sigma-1}$ are linearly independent with respect to the set $\mathcal{V}_0 := \{f \cdot p(z) : p(z) \in \mathbb{Q}[z]^m\}$ (considered as a vector space over $\mathbb{Q}$), in general a proper subset of $\mathcal{V}$. In terms of linear algebra, this is equivalent to say that the data $(\mathbf{C}_\sigma, \mathbf{F}_\sigma)$ are *controllable*, i.e., for large k, the columns of $\mathbf{F}_\sigma, \mathbf{C}_\sigma \cdot \mathbf{F}_\sigma, \ldots, \mathbf{C}_\sigma^k \cdot \mathbf{F}_\sigma$ generate the whole space $\mathbb{Q}^\sigma$. Moreover, from system theory (see, e.g., [36, p.426ff,p.481ff]) it is well–known that this necessary condition is also sufficient for the existence of a normal point $\vec{n}$ with $|\vec{n}| = \sigma$.

We will say that $(\mathbf{C}, \mathbf{F})$ is controllable if $(\mathbf{C}_\sigma, \mathbf{F}_\sigma)$ are controllable for all $\sigma \geq 0$. One easily verifies the equivalent condition that, for each $\sigma \geq 0$, there exists an element of $\mathcal{V}_0$ having exact order σ. Such a regularity assumption has been imposed for several algorithms for solving the approximation problems mentioned in the examples of Section 2. Also, equivalent characterizations have been established: in the case of M–Padé approximation (see Example 2.5), it is shown in [5, Lemma 3.1] that $(\mathbf{C}, \mathbf{F})$ is controllable iff the vector of functions $f = [f_1, .., f_m]$ does not vanish identically at any of the involved

knots. In particular, for Hermite–Padé approximation we have the equivalent requirement $f(0) \neq 0$. Moreover [9, Lemma 2.7], for vector Hermite–Padé approximants (see Example 2.3), $(\mathbf{C}, \mathbf{F})$ is controllable iff the $s \times m$ matrix $f(0)$ has maximal rank.

Though such a condition allows us to simplify notation, for an application of our theory to the matrix–GCD problem we need to also allow for non–controllable $(\mathbf{C}, \mathbf{F})$. One possibility to remedy this drawback is to introduce additional functions $f^{(m+1)}, f^{(m+2)}, ...$, and thus to consider a suitable extension $\mathbf{F}^*$ of $\mathbf{F}$. Instead, we prefer to consider a particular maximal subsequence of linear functionals being linearly independent with respect to $\mathcal{V}_0$. The symbol $*$ will be used to identify the resulting Krylov–matrices and multigradients.

We define a unique sequence of integers $(\sigma(j))_{j=0,1,..}$ being the indices of our maximal subsequence of linearly independent linear functionals by the following requirements: for all non–negative integers j there holds

$$c_{\sigma(0)}, c_{\sigma(1)}, ..., c_{\sigma(j)} \text{ are linearly independent w. r. t. } \mathcal{V}_0, \tag{9}$$

$$c_{\sigma(0)}, ..., c_{\sigma(j-1)}, c_\sigma \text{ are linearly dependent w. r. t. } \mathcal{V}_0 \text{ for all } 0 \leq \sigma < \sigma(j). \tag{10}$$

Definition 4.2 (Para–normality, σ–Normality)
Let $\vec{n}$ be a multi–index, and let j, σ be non–negative integers. We denote by $\mathbf{K}^*(\vec{n}, j)$ the matrix of size $j \times |\vec{n}|$ obtained by taking the rows labeled $\sigma(0), ..., \sigma(j-1)$ of the ordinary striped Krylov matrix $\mathbf{K}(\vec{n}, \sigma(j))$. The scalar

$$d^*(\vec{n}) = det(\mathbf{K}^*(\vec{n}, |\vec{n}|))$$

will be referred to as the modified multigradient *of $\mathbf{F}$ of type $\vec{n}$. The multi-index $\vec{n}$ is called* para–normal *if $d^*(\vec{n}) \neq 0$, and called* σ–normal *if it is para–normal and $\sigma(|\vec{n}| - 1) < \sigma \leq \sigma(|\vec{n}|)$ (where $\sigma(-1) := -1$).* □

Note that the concepts of para–normality and of normality (in the sense of Definition 4.1) coincide exactly in the case of controllable $(\mathbf{C}, \mathbf{F})$. Moreover, $\vec{n}$ is $|\vec{n}|$–normal iff it is a normal point. This implies in particular that $\sigma(j) = j$ for $j = 0, 1, ..., |\vec{n}| - 1$, that is, $(\mathbf{C}_{|\vec{n}|}, \mathbf{F}_{|\vec{n}|})$ is controllable. Also, by exploiting the dependency relations (10) one gets a special multiplication rule of the form (2) connecting only the linearly independent linear functionals

$$c_{\sigma(j)}(z \cdot f) = c^*_{j,0} \cdot c_{\sigma(0)}(f) + \cdots + c^*_{j,j} \cdot c_{\sigma(j)}(f)$$

for all $f \in \mathcal{V}_0$ and for all $j \geq 0$, with $c^*_{j,k} \in \mathbb{Q}$. Hence modified striped Krylov matrices $\mathbf{K}^*(\vec{n}, j)$ may be represented themselves as striped Krylov matrices with controllable data $(\mathbf{C}^*, \mathbf{F}^*)$. However, in the sequel we will not make use of this result. A final characterization is mentioned in the following

Lemma 4.3 *The multi-index $\vec{n}$ is σ–normal if and only if any striped Krylov matrix $\mathbf{K}(\vec{n}', \sigma)$ containing the submatrix $\mathbf{K}(\vec{n}, \sigma)$ has rank $|\vec{n}|$. In this case, a maximal invertible submatrix is given by $\mathbf{K}^*(\vec{n}, |\vec{n}|)$.*

Proof: Apply Gaussian elimination with column pivoting to $\mathbf{K}(\vec{n}', \sigma)$. □

5 Mahler Systems

In this section we introduce the notion of a Mahler System. These systems are generalizations of the Padé and matrix-type Padé systems of [18, 39, 41] and, up to a constant factor, have already been considered by Mahler [45] in the case of perfect systems for Hermite-Padé and simultaneous Padé approximants. They are also the fundamental building blocks that we use for the fraction-free algorithm presented in the later sections.

For a given multi-index $\vec{n}$ define $r(\vec{n}, z)$ and $p^{(\ell)}(\vec{n}, z)$ by $r(\vec{0}, z) = 0$, $p^{(\ell)}(\vec{n}, z) = 0$ in the case $\vec{n}^{(\ell)} = 0$, and otherwise by the determinant formulas

$$r(\vec{n}, z) = det \begin{bmatrix} \mathbf{K}^*(\vec{n}, |\vec{n}| - 1) \\ \hline \mathbf{E}(z) \end{bmatrix}$$

where

$$\mathbf{E}(z) = [f^{(1)}, \ldots, z^{\vec{n}^{(1)}-1} f^{(1)} | \ldots\ldots | f^{(m)}, \ldots, z^{\vec{n}^{(m)}-1} f^{(m)}],$$

and

$$p^{(\ell)}(\vec{n}, z) = det \begin{bmatrix} \mathbf{K}^*(\vec{n}, |\vec{n}| - 1) \\ \hline \mathbf{E}^{(\ell)}(z) \end{bmatrix}$$

with

$$\mathbf{E}^{(\ell)}(z) = \mathbf{E}^{(\ell)}(\vec{n}, z) = [0, \ldots, 0 | 1, z, \ldots, z^{\vec{n}^{(\ell)}-1} | 0, \ldots, 0]. \tag{11}$$

The nonzero entries in $\mathbf{E}^{(\ell)}(z)$ occur in the ℓ-th stripe. We also let $p(\vec{n}, z) = [p^{(1)}(\vec{n}, z), \ldots, p^{(m)}(\vec{n}, z)]^T$ be the column vector of polynomials defined above.

Lemma 5.1 *For a multi-index $\vec{n}$ we have*
(a) $f \cdot p(\vec{n}, z) = r(\vec{n}, z) \in \mathcal{V}_0$.

(b) $\mathrm{ord}\,(r(\vec{n},z)) \geq \sigma(|\vec{n}|-1)$ *and* $c_{\sigma(|\vec{n}|-1)}(r(\vec{n},z)) = d^*(\vec{n})$.
(c) $deg(p^{(\ell)}(\vec{n},z)) \leq \vec{n}^{(\ell)} - 1$. *Moreover, if* $\vec{n}^{(\ell)} > 0$ *then the* $\vec{n}^{(\ell)}-1$*-st coefficient is*

$$p^{(\ell)}_{\vec{n}^{(\ell)}-1} = \epsilon^{(\ell)}(\vec{n}) \cdot d^*(\vec{n} - \vec{e}_\ell), \quad \epsilon^{(\ell)}(\vec{n}) := (-1)^{\vec{n}^{(\ell+1)}+\ldots+\vec{n}^{(m)}}.$$

(d) $p(\vec{n},z)$ *is not identically zero if and only if, up to multiplication by a scalar from* $\mathbb{Q}$*, there exists exactly one solution of type* $(\sigma(|\vec{n}|-2)+1, \vec{n})$ *(being given by* $p(\vec{n},z)$*).*

Proof: Part (a) follows by expanding determinants with respect to the last row. In order to show part (b) notice that, for $i = \sigma(j)$, $0 \leq j < |\vec{n}|-1$, $c_i(r(\vec{n},z))$ is a determinant of a matrix with two equal rows and hence is zero. In the case $i \in \{0,\ldots,\sigma(|\vec{n}|-1)-1\} \setminus \{\sigma(0),\ldots,\sigma(|\vec{n}|-2)\}$ we obtain $c_i(r(\vec{n},z)) = 0$ according to (10). The first potential case where a possibly linearly independent row occurs is when $i = \sigma(|\vec{n}|-1)$, and thus $c_i(r(\vec{n},z)) = d^*(\vec{n})$. Part (c) follows by expanding out the determinant definition of $p^{(\ell)}(\vec{n},z)$ along the last row. The coefficient is, at least up to sign, the same as taking determinants of the matrix determined by eliminating the last row and column $\vec{n}^{(1)} + \cdots + \vec{n}^{(\ell)}$, which is just $d^*(\vec{n} - \vec{e}_\ell)$. The sign is determined by counting the number of columns from the bottom right corner of the matrix. Finally, the assertion of part (d) is a consequence of Cramer's rule applied to the homogeneous system of linear equations $\mathbf{K}^*(\vec{n}, |\vec{n}|-1) \cdot \mathbf{P} = 0$, since in fact $p(\vec{n},z) \not\equiv 0$ if and only if the rank of the matrix $\mathbf{K}^*(\vec{n}, |\vec{n}|-1)$ of size $(|\vec{n}|-1) \times |\vec{n}|$ is maximal. □

Lemma 5.1 says that $p(\vec{n},z)$ is a solution in $\mathbb{D}^m[z]$ to our interpolation problem of Definition 2.1 of type $(\sigma,\vec{n})$, $\sigma \leq \sigma(|\vec{n}|-1)$. However, one rarely wants to use this definition in order to compute this solution. Rather it is better to use systems of linear equations for this computation. For instance, suppose that $\vec{n}$ is a normal point. Then solving the system (6) using Cramer's rule over $\mathbb{Q}$ gives a solution $\mathbf{P}$ of problem (7) with $\mathbf{P}$ being a vector having only coefficients from $\mathbb{D}$. From Lemma 5.1 (b),(d) one sees that $\mathbf{P}$ provides the coefficients of the polynomials $p(\vec{n},z)$ via partitioning the coefficient vector as

$$\mathbf{P} = [p^{(1)}_0, \ldots, p^{(1)}_{\vec{n}^{(1)}-1} | \ldots\ldots | p^{(m)}_0, \ldots, p^{(m)}_{\vec{n}^{(m)}-1}].$$

Similarly, suppose that $\vec{n}$ is para–normal (see Definition 4.2) and choose σ such that $\vec{n}$ is σ–normal. By Lemma 4.3 we have *rank* $\mathbf{K}(\vec{n},\sigma)$ = *rank* $\mathbf{K}(\vec{n}+\vec{e}_i,\sigma) = |\vec{n}|$ for all $i = 1,\ldots,m$, with a square submatrix of maximal rank being given by $\mathbf{K}^*(\vec{n},|\vec{n}|)$. Therefore we may find unique solutions for the systems of equations (usually referred to as *fundamental equations* [35] or *Yule–Walker equations* of the corresponding striped Krylov matrix)

$$\mathbf{K}(\vec{n},\sigma) \cdot \bar{\mathbf{P}}^{(i)} = -\mathbf{C}^{\vec{n}^{(i)}}_\sigma \cdot \mathbf{F}^{(i)}_\sigma, \quad 1 \leq i \leq m.$$

Again using Cramer's rule (with respect to $\mathbf{K}^*(\vec{n},|\vec{n}|)$), we obtain solutions $\tilde{\mathbf{P}}^{(i)}$ of elements from the domain $\mathbb{D}$ to the systems

$$\mathbf{K}(\vec{n},\sigma) \cdot \tilde{\mathbf{P}}^{(i)} = -d^*(\vec{n}) \cdot \mathbf{C}^{\vec{n}^{(i)}}_\sigma \cdot \mathbf{F}^{(i)}_\sigma, \quad 1 \leq i \leq m. \tag{12}$$

Thus, by part (c) of Lemma 5.1, the vector $\tilde{\mathbf{P}}^{(i)}$ consists of the coefficients of the vector of determinant polynomials $\epsilon^{(i)}(\vec{n}) \cdot p(\vec{n} + \vec{e}_i, z)$.

We are interested in recursively or iteratively computing solutions of equation (3). However to do this we need a larger collection of solutions to the problem. One can think of the scalar GCD problem as an example - there one needs two remainders at every step to get the next remainder. In our case we need to look for the m solutions described already by (12).

Definition 5.2 (Mahler Systems)
The $m \times m$ matrix of polynomials

$$\mathbf{M}(\vec{n}, z) = [\mathbf{M}^{(\lambda,j)}(\vec{n}, z)]_{\lambda,j=1}^{m}, \quad \mathbf{M}^{(\lambda,j)}(\vec{n}, z) := \epsilon^{(j)}(\vec{n}) \cdot p^{(\lambda)}(\vec{n} + \vec{e}_j, z),$$

is called the Mahler System *of type $\vec{n}$. We denote its j-th column by $\mathbf{M}^{(\cdot,j)}(\vec{n}, z)$.* □

Some Mahler systems for Hermite–Padé approximation may be found in Example 6.2 below. For the particular case of M–Padé approximation at a normal point $\vec{n}$, our Mahler system coincide with that proposed by Mahler [45] (up to the common scalar factor $d^*(\vec{n})$). In the sequel, we will only consider Mahler systems at para–normal points for which we may establish several equivalent characterizations

Lemma 5.3 *Let $\vec{n}$ be a multi-index, and $\lambda \in \{1, .., m\}$. The following assertions are pairwise equivalent:*
(a) $\vec{n}$ is a para–normal point.
(b) $\deg p^{(\lambda)}(\vec{n} + \vec{e}_\lambda, z) = \vec{n}^{(\lambda)}$.
(c) A solution of type $(\sigma(|\vec{n}|-1)+1, \vec{n}+\vec{e}_\lambda)$ is unique up to multiplication with an element from $\mathbb{Q}$, with its λ-th component having exact degree $\vec{n}^{(\lambda)}$.
(d) For any $\sigma > \sigma(|\vec{n}| - 1)$, a solution of type $(\sigma, \vec{n})$ is necessarily trivial.
(e) The columns of the Mahler system $\mathbf{M}(\vec{n}, z)$ are linearly independent over $\mathbb{Q}[z]$.

Proof: The equivalence of assertion (a) and any of the assertions (b) or (c) follows from Lemma 5.1 and the following remarks. In order to establish equivalence between (a) and (d), notice that the coefficient vector $\mathbf{P}$ of a solution $p(z)$ of type $(\sigma(|\vec{n}| - 1) + 1, \vec{n})$ necessarily satisfies $\mathbf{K}(\vec{n}, \sigma(|\vec{n}| - 1) + 1) \cdot \mathbf{P} = 0$. By definition (9), (10), we obtain the equivalent system of equations $\mathbf{K}^*(\vec{n}, |\vec{n}|) \cdot \mathbf{P} = 0$, with a square matrix of coefficients. Thus $\mathbf{K}^*(\vec{n}, |\vec{n}|)$ is nonsingular or, in other words, $d^*(\vec{n}) \neq 0$ if and only if each such solution $\mathbf{P}$ is trivial.

For the equivalence between (a) and (e) it is sufficient to show that $det\,\mathbf{M}(\vec{n},z) \neq 0$ if and only if $d^*(\vec{n}) \neq 0$. Notice that the elements of $\mathbf{M}(\vec{n},z)$, namely, $\mathbf{M}^{(\lambda,j)}(\vec{n},z)$, $\lambda,j = 1,..,m$, are determinants of matrices of size $(|\vec{n}|+1)\times(|\vec{n}|+1)$. These matrices are obtained by bordering the matrix $\mathbf{K}^*(\vec{n},|\vec{n}|)$ on the bottom by one additional row and on the right by one additional column. Let $\vec{e} := (1,1,...,1)$, and let $\mathbf{E}^{(\lambda)}(\vec{n},z)$ be defined as in (11). Then, by the Sylvester determinantal identity, we have

$$det\,\mathbf{M}(\vec{n},z) = (det\,\mathbf{K}^*(\vec{n},|\vec{n}|))^{m-1}\cdot\beta(z),$$

where $\beta(z)$ denotes the determinant of the augmented matrix

$$\beta(z) = \pm det\left[\begin{array}{c} \mathbf{K}^*(\vec{n}+\vec{e},|\vec{n}|) \\ \hline \mathbf{E}^{(1)}(\vec{n}+\vec{e},z) \\ \vdots \\ \mathbf{E}^{(m)}(\vec{n}+\vec{e},z) \end{array}\right].$$

Expanding $\beta(z)$ with respect to the last m rows shows that $\beta(z)$ is a polynomial in z, and that more precisely [2]

$$\beta(z) = \pm d^*(\vec{n})\cdot z^{|\vec{n}|} + \alpha(z), \quad deg\,\alpha < |\vec{n}|.$$

Here we have taken into account that the coefficient of $z^{|\vec{n}|}$ in $\beta(z)$ is obtained by the cofactor of $diag(z^{\vec{n}^{(1)}},...,z^{\vec{n}^{(m)}})$ in $\beta(z)$. Consequently, $det\,\mathbf{M}(\vec{n},z) = \pm d^*(\vec{n})^{m-1}\cdot(d^*(\vec{n})\cdot z^{|\vec{n}|}\pm\alpha)$. Therefore the two quantities $det\,\mathbf{M}(\vec{n},z)$ and $d^*(\vec{n})$ only simultaneously become zero. □

Given a para–normal multi–index $\vec{n}$, we will mostly apply Lemma 5.3 in order to verify that a given matrix polynomial is a Mahler system of type $\vec{n}$. Here we just have to check that, for $\lambda = 1,..,m$, the λ-th column is a solution of type $(\sigma(|\vec{n}|-1)+1,\vec{n}+\vec{e}_\lambda)$ with the correct normalization, i.e., the coefficient of $z^{\vec{n}^{(\lambda)}}$ of the λ-th component equals $d^*(\vec{n})$.

To the end of this section, we state a further equivalent characterization of para–normal multi–indices. This statement will be proved at the end of Section 7 where additional results are available. For the remainder of this paper we will use the abbreviation $z^{\vec{\nu}}$ for denoting the diagonal matrix $diag(z^{\vec{\nu}^{(1)}},...,z^{\vec{\nu}^{(m)}})$.

[2] One shows that, for controllable $(\mathbf{C},\mathbf{F})$,

$$det\,\mathbf{M}(\vec{n},z) = \pm d^*(\vec{n})^m\cdot\prod_{k=0}^{|\vec{n}|-1}(z-c_{k,k})$$

(for the approximation problems of Section 2, see [46, p.42], [3, p.90-91], or [9, Lemma 2.7]).

Corollary 5.4 *Let $\vec{n}$ be a multi–index, and $\sigma > \sigma(|\vec{n}|-1)$. Then $\vec{n}$ is σ–normal iff there exists a matrix polynomial $\mathbf{M}(z)$ with columns having order $\geq \sigma$ which satisfies the degree constraints*

$$z^{-\vec{n}} \cdot \mathbf{M}(z) = c \cdot \mathbf{I}_m + \mathcal{O}(z^{-1})_{z\to\infty}, \quad c \in \mathbb{Q} \setminus \{0\}.$$

In this case, $\mathbf{M}(z) = \frac{c}{d^*(\vec{n})} \cdot \mathbf{M}(\vec{n}, z)$.

6 Computing Mahler Systems along Perfect Paths

For a given multi-index $\vec{n}$, we are interested in computing a solution of type $(|\vec{n}|-1, \vec{n})$ to the interpolation problem of Definition 2.1 in a fraction-free way. By Lemma 5.1, the polynomial vector $p(\vec{n}, z)$ defined in the previous section provides one solution to this problem. Of course, to compute these polynomials one does not want to use the determinant definition, except perhaps for small problems. In this section we give a fast method to compute the solution to our rational interpolation problem using only polynomial operations over the integral domain $\mathbb{D}$. However, for the algorithm presented in this section we require some regularity assumptions, which are no longer necessary for the algorithm presented in the next section.

In the case where we are at a normal point $\vec{n}$ the next theorem tells us (in a more general setting) how to compute a Mahler system at a subsequent normal point $\vec{n}+\vec{e}_\lambda$ from the Mahler system at $\vec{n}$. A similar recurrence relation for Hermite–Padé approximation has been established earlier by Paszkowski [46, 47, 48], and generalized by one of the authors [3, Kapitel 3.3], however, without noticing that this is the key for fraction–free computations.

Theorem 6.1 *Suppose that $\vec{n}$ is para–normal. Furthermore, let $\sigma(|\vec{n}|-1) < \sigma \leq \sigma(|\vec{n}|)$, and for $\ell = 1, .., m$ set*

$$r^{(\ell)} := c_\sigma(f \cdot \mathbf{M}^{(\cdot,\ell)}(\vec{n}, z)).$$

(a) *$\vec{n}$ is also $(\sigma+1)$–normal (i.e., $\sigma < \sigma(|\vec{n}|)$) iff $r^{(1)} = r^{(2)} = .. = r^{(m)} = 0$.*
(b) *$\vec{n} + \vec{e}_\lambda$ is a para–normal point iff $r^{(\lambda)} \neq 0$.*
(c) *In the case $r^{(\lambda)} \neq 0$, we define also for $\ell = 1, .., m$, $\ell \neq \lambda$*

$$p^{(\ell)} := \text{coefficient}(\mathbf{M}^{(\ell,\lambda)}(\vec{n}, z), z^{\vec{n}^{(\ell)}-1}).$$

Then $\mathbf{M}(\vec{n} + \vec{e}_\lambda, z)$ can be computed from $\mathbf{M}(\vec{n}, z)$ as follows

$$\mathbf{M}^{(\cdot,\ell)}(\vec{n} + \vec{e}_\lambda, z) \cdot p^{(\lambda)} \cdot \epsilon^{(\lambda)}(\vec{n}) = \mathbf{M}^{(\cdot,\ell)}(\vec{n}, z) \cdot r^{(\lambda)} - \mathbf{M}^{(\cdot,\lambda)}(\vec{n}, z) \cdot r^{(\ell)} \tag{13}$$

for $\ell = 1, 2, ..., m$, $\ell \neq \lambda$, and

$$\begin{aligned} \mathbf{M}^{(\cdot,\lambda)}(\vec{n} + \vec{e}_\lambda, z) \cdot p^{(\lambda)} \cdot \epsilon^{(\lambda)}(\vec{n}) \;=\; & (z - c_{\sigma,\sigma}) \cdot \mathbf{M}^{(\cdot,\lambda)}(\vec{n}, z) \cdot r^{(\lambda)} \\ & - \sum_{\ell\neq\lambda} \mathbf{M}^{(\cdot,\ell)}(\vec{n} + \vec{e}_\lambda, z) \cdot p^{(\ell)} \cdot \epsilon^{(\lambda)}(\vec{n}). \end{aligned} \tag{14}$$

Proof: For a proof of part (a), set

$$B := \mathbf{K}(\vec{n} + [\sigma+1, \sigma+1, .., \sigma+1], \sigma), \quad B' := \mathbf{K}(\vec{n} + [\sigma+1, \sigma+1, .., \sigma+1], \sigma+1).$$

By Lemma 4.3 along with our assumptions, we have that *rank* $B = |\vec{n}|$, and from the Cayley–Hamilton theorem we know that *rank* $B' \geq$ *rank* $\mathbf{K}(\vec{n}', \sigma+1)$ for any multi–index $\vec{n}'$. Hence from definition (9), (10) we obtain the characterization $\sigma < \sigma(|\vec{n}|)$ iff *rank* $B =$ *rank* B'. The $m \cdot (\sigma+1)$ coefficient vectors of the polynomial vectors

$$(z - c_{\sigma,\sigma})^j \cdot \mathbf{M}^{(\cdot,\ell)}(\vec{n}, z), \quad \ell = 1, .., m, \quad j = 0, ..., \sigma,$$

are easily shown to be elements of the kernel of B, and are linearly independent over $\mathbb{Q}$ by Lemma 5.3(e). Thus we have found a basis of the kernel of B. Notice also that, according to (2), the order of $f \cdot (z - c_{\sigma,\sigma})^j \cdot \mathbf{M}^{(\cdot,\ell)}(\vec{n}, z)$ is larger than σ if $j > 0$. As a consequence, we have established the equivalencies $\sigma < \sigma(|\vec{n}|)$ iff the kernels of B and B' coincide iff $f \cdot \mathbf{M}^{(\cdot,\ell)}(\vec{n}, z)$ has order $\geq \sigma+1$ for $\ell = 1, .., m$, as claimed in part (a).

Assertion (b) follows from part (a) together with Lemma 5.1(b).

In order to show the recurrence relation (13) for the case $\ell \neq \lambda$, let

$$q(z) := \mathbf{M}^{(\cdot,\ell)}(\vec{n} + \vec{e}_\lambda, z) \cdot p^{(\lambda)} \cdot \epsilon^{(\lambda)}(\vec{n}) - \mathbf{M}^{(\cdot,\ell)}(\vec{n}, z) \cdot r^{(\lambda)} + \mathbf{M}^{(\cdot,\lambda)}(\vec{n}, z) \cdot r^{(\ell)}.$$

We claim that $q(z) = 0$. First by construction we get $ord(f \cdot q(z)) \geq \sigma+1$. Furthermore, $deg\, q^{(\mu)}(z) \leq \vec{n}^{(\mu)} - 1 + \delta_{\mu,\ell} + \delta_{\mu,\lambda}$. More precisely, the coefficient of $z^{\vec{n}^{(\ell)}}$ of the ℓ-th component of $q(z)$ is given by

$$d^*(\vec{n} + \vec{e}_\lambda) \cdot p^{(\lambda)} \cdot \epsilon^{(\lambda)}(\vec{n}) - d^*(\vec{n}) \cdot r^{(\lambda)} = 0$$

since $p^{(\lambda)} = d^*(\vec{n})$ due to Lemma 5.1(c), and $r^{(\lambda)} = \epsilon^{(\lambda)}(\vec{n}) \cdot d^*(\vec{n} + \vec{e}_\ell)$ due to Lemma 5.1(b). Hence $q(z)$ is a solution of type $(\vec{n} + \vec{e}_\lambda, \sigma+1)$, and thus by Lemma 5.3(d) is identically zero.

Identity (14) is shown in a similar manner, let

$$q(z) := (z - c_{\sigma,\sigma}) \cdot \mathbf{M}^{(\cdot,\lambda)}(\vec{n}, z) \cdot d^*(\vec{n} + \vec{e}_\lambda) - \sum_{\ell=1}^{m} \mathbf{M}^{(\cdot,\ell)}(\vec{n} + \vec{e}_\lambda, z) \cdot p^{(\ell)}.$$

Since $d^*(\vec{n} + \vec{e}_\lambda) = r^{(\lambda)} \cdot \epsilon^{(\lambda)}(\vec{n})$, it is sufficient to prove that $q(z)$ vanishes identically, which follows again by Lemma 5.3(d) by checking order and degree of $q(z)$. First notice that $ord(f \cdot (z - c_{\sigma,\sigma}) \cdot \mathbf{M}^{(\cdot,\lambda)}(\vec{n}, z)) \geq \sigma+1$ by (2). Moreover, all terms in the sum have order at least $\sigma+1$, and thus $ord(f \cdot q(z)) \geq \sigma+1$. Also, by definition, the μ-th component of $q(z)$ contains only powers z^j with $j = 0, 1, ..., \vec{n}^{(\mu)} + \delta_{\lambda,\mu} =: j_\mu$. By using Lemma 5.1 (c), one verifies that the factors in the sum have been chosen such that the coefficient before z^{j_μ} in $q^{(\mu)}(z)$ vanishes, and hence $deg\, q^{(\mu)}(z) \leq \vec{n}^{(\mu)} - 1 + \delta_{\mu,\lambda}$ for all μ. Thus $q(z) = 0$. □

Table 1: *The algorithm FFFGnormal*

ALGORITHM FFFGnormal (on arbitrary staircases consisting of normal points)

INPUT: a vector of formal series f,
a staircase $(\vec{n}_k)_{k=0,..,K}$ of normal points.

OUTPUT: For $k = 0, 1, 2, \ldots, K$ with $\epsilon_k \in \{-1, 1\}$:
Mahler systems $\mathbf{M}_k = \epsilon_k \cdot \mathbf{M}(\vec{n}_k, z)$,
multigradients $d_k = \epsilon_k \cdot d^*(\vec{n}_k)$.

INITIALIZATION: $\mathbf{M}_0 \leftarrow \mathbf{I}_m$, $d_0 \leftarrow 1$

ITERATIVE STEP: For $k = 0, 1, 2, .., K-1$:
Define $\lambda \in \{1, .., m\}$ by $\vec{n}_{k+1} - \vec{n}_k = \vec{e}_\lambda$.

Calculate for $\ell = 1, .., m$:
first term of residuals $r^{(\ell)} \leftarrow c_k(f \cdot \mathbf{M}_k^{(\cdot,\ell)})$,
leading coefficients $p^{(\ell)} \leftarrow \mathit{coefficient}(\mathbf{M}_k^{(\ell,\lambda)}, z^{\vec{n}_k^{(\ell)}-1})$.

Increase order for $\ell = 1, .., m$, $\ell \neq \lambda$:
$\mathbf{M}_{k+1}^{(\cdot,\ell)} \leftarrow [\mathbf{M}_k^{(\cdot,\ell)} \cdot r^{(\lambda)} - \mathbf{M}_k^{(\cdot,\lambda)} \cdot r^{(\ell)}]/d_k$
$\mathbf{M}_{k+1}^{(\cdot,\lambda)} \leftarrow (z - c_{k,k}) \cdot \mathbf{M}_k^{(\cdot,\lambda)}$

Adjust degree constraints:
$\mathbf{M}_{k+1}^{(\cdot,\lambda)} \leftarrow [\mathbf{M}_{k+1}^{(\cdot,\lambda)} \cdot r^{(\lambda)} - \sum_{\ell \neq \lambda} \mathbf{M}_{k+1}^{(\cdot,\ell)} \cdot p^{(\ell)}]/d_k$

New multigradient: $d_{k+1} = r^{(\lambda)}$

Theorem 6.1 leads to an algorithm to compute solutions to the rational interpolation problem on staircases under the assumption that all intermediate problems are at normal points. Here we denote by staircase a sequence $(\vec{n}_k)_{k=0,1,\ldots}$ of multi-indices with the properties that

$$\vec{n}_0 = \vec{0},\ \vec{n}_{|\vec{n}|} = \vec{n}, \text{ and } \forall k \geq 0\, \exists \lambda_k \text{ such that } \vec{n}_{k+1} - \vec{n}_k = \vec{e}_{\lambda_k}. \tag{15}$$

At every step $\vec{n}_k$ we find a λ such that $\vec{n}_{k+1} = \vec{n}_k + \vec{e}_\lambda$ is normal (which is for instance the case when the vector f is perfect, see Definition 4.1). Then, using the iteration given by Theorem 6.1 with $\sigma = |\vec{n}_k| = k$, we see that we can remove the scalar common factor $p^{(\mu)} = d^*(\vec{n}_k)$ before we proceed with our next iteration. This scalar is determined as the leading coefficient of the (λ, λ) term of the k-th Mahler system.

Therefore, not only the representations (7), (12) of the solutions, but also recurrence (13) reminds one of the well–known recurrence relations of fraction–free Gaussian elimination [2, 29]. On the other hand, relation (14) gives a significant gain in complexity in

comparison with the classical Gaussian elimination, obtained by taking into account the particular structure of our block Krylov matrices. This serves as motivation to refer to our algorithm proposed in Table 1 as **F**raction–**F**ree **F**ast **G**aussian elimination.

From Theorem 6.1 one can see that the iteration is best done in two stages. If we have the Mahler system of type $\vec{n}_k$ and wish to compute the Mahler system of type $\vec{n}_{k+1} = \vec{n}_k + \vec{e}_{\lambda_k}$ then we first increase the order of all the columns of $\mathbf{M}(\vec{n}_k, z)$. This is done by using column λ_k to increase the orders of all the other columns using (13) of Theorem 6.1. The λ_k-th column itself has its order increased by multiplication by $z - c_{|\vec{n}_k|,|\vec{n}_k|}$. At this stage all the columns except λ_k are constant multiples of the corresponding columns of $\mathbf{M}(\vec{n}_k + \vec{e}_{\lambda_k}, z)$. We pull out the constant from these columns to make them the same as the corresponding columns of the new Mahler system. Finally, column λ_k does not have the correct degree structure as required for our new Mahler system. We then use all the other columns to return this degree structure to the desired form. This gives column λ_k as a constant multiple of the λ_k-th column of $\mathbf{M}(\vec{n}_k + \vec{e}_{\lambda_k}, z)$. Removing this constant gives the correct λ_k-th column of $\mathbf{M}(\vec{n}_k + \vec{e}_{\lambda_k}, z)$ and hence the new Mahler system.

In the algorithm FFFGnormal stated in Table 1, one may find a slight simplification of relations (13), (14). In fact, we prefer to compute Mahler systems only up to sign, namely $\mathbf{M}_k = \epsilon_k \cdot \mathbf{M}(\vec{n}_k, z)$ with

$$\epsilon_0 = 1, \quad \epsilon_{k+1} = \epsilon^{(\lambda_k)}(\vec{n}_k) \cdot \epsilon_k, \quad k \geq 0, \tag{16}$$

since then all terms $\epsilon^{(\lambda_k)}(\vec{n}_k)$ in (13), (14) may be dropped.

In Table 1, we have not discussed in detail how to compute efficiently the first term of the residuals, namely $r^{(\ell)}$, $\ell = 1, .., m$. One possibility (mainly applicable for Hermite–Padé approximation and its vector counterpart) is to compute explicitly $c_k(f \cdot \mathbf{M}_\sigma^{(\cdot,\ell)})$ by determining a particular coefficient of the scalar product $f \cdot \mathbf{M}_\sigma$. Another approach, which may be preferable for more complicated special multiplication rules (2), is to simultaneously compute all required values of the residuals, i.e., to compute the (non–trivial part of the) *residual vectors*

$$\mathbf{R}_k^{(\ell)} = [c_\sigma(f \cdot \mathbf{M}_k^{(\cdot,\ell)})]_{\sigma=0,..,K-1}.$$

Here we use the initializations $\mathbf{R}_0^{(\ell)} = \mathbf{F}^{(\ell)}$, and obtain according to Table 1 and (2) the recurrences

$$\mathbf{R}_{k+1}^{(\ell)} = \begin{cases} [\mathbf{R}_k^{(\ell)} \cdot r^{(\lambda)} - \mathbf{R}_k^{(\lambda)} \cdot r^{(\ell)}]/d_k & \text{for } \ell \neq \lambda, \\ [(\mathbf{C}_K - c_{k,k} \cdot \mathbf{I}_K) \cdot \mathbf{R}_k^{(\lambda)} \cdot r^{(\lambda)} - \sum_{\ell \neq \lambda} \mathbf{R}_{k+1}^{(\ell)} \cdot p^{(\ell)}]/d_k & \text{for } \ell = \lambda. \end{cases}$$

We again observe close relationships to the recurrence relations of the classical one–step fraction–free Gaussian elimination [2, 29]. We also mention that multi–step elimination schemes may be given. However, due to our special rule, the formalism becomes more complicated.

Example 6.2
Let f be the vector of power series[3] whose first 6 terms are

$$\left[1 - z + 19\,z^2 + 3\,z^3 - 5\,z^5,\ \ 9 + 6\,z - 5\,z^2 + 5\,z^3 + 4\,z^5,\ \ 1 + 9\,z^2 + 9\,z^3 - 4\,z^5\right].$$

Then the Mahler systems of f of type $[1,0,0]$, $[1,1,0]$, $[1,1,1]$ and $[2,1,1]$ generated by the preceding algorithm are given by

$$\mathbf{M}_1 = \mathbf{M}(\vec{n}_1, z) = \begin{bmatrix} z & -9 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{M}_2 = \mathbf{M}(\vec{n}_2, z) = \begin{bmatrix} 15\,z + 9 & 81 & -6 \\ -1 & 15\,z - 9 & -1 \\ 0 & 0 & 15 \end{bmatrix},$$

$$\mathbf{M}_3 = \mathbf{M}(\vec{n}_3, z) = \begin{bmatrix} 26\,z + 80 & 810 & 86 \\ 9 & 26\,z + 96 & 10 \\ -161 & -1674 & 26\,z - 176 \end{bmatrix},$$

and

$$\mathbf{M}_4 = \mathbf{M}(\vec{n}_4, z) = \begin{bmatrix} -670\,z^2 + 138\,z + 270 & 12286\,z + 16930 & 1042\,z + 990 \\ 22 & -670\,z + 1779 & 103 \\ -468 & -32941 & -670\,z - 1917 \end{bmatrix}.$$

The residuals determined by $f \cdot \mathbf{M}_4$ are given by

$$\left[\ -12316\,z^4 + O\left(z^5\right),\ \ 33508\,z^4 + O\left(z^5\right),\ \ -2904\,z^4 + O\left(z^5\right)\ \right]$$

so that in this step $r^{(1)} = d([3,1,1]) = -12316$, $r^{(2)} = -d([2,2,1]) = 33508$ and $r^{(3)} = d([2,1,2]) = -2904$ (see Lemma 5.1(b)). Also, the leading coefficients of the polynomials on the diagonal of the Mahler system $\mathbf{M}_4$ are equal to $d_4 = d([2,1,1]) = -670$. In order to generate the Mahler system $\mathbf{M}_5 = -\mathbf{M}(\vec{n}_5, z)$ of type $[2,2,1]$, the algorithm first increases the orders of all the columns by combining column ℓ with column 2, $\ell = 1, 3$, and by multiplying the second column by z. This gives

$$\tilde{\mathbf{P}} = \begin{bmatrix} 33508\,z^2 - 232744\,z - 324712 & 12286\,z^2 + 16930z & -105364\,z - 122892 \\ 12316\,z - 33802 & -670\,z^2 + 1779z & 2904\,z - 12862 \\ 628930 & -32941z & 33508\,z + 238650 \end{bmatrix}.$$

Note that the common multiplier $d_4 = -670$ has been removed from the computations of columns 1 and 3. The algorithm then uses the values $p^{(1)} = 12286 = d([1,2,1])$ with the first column and $p^{(3)} = -32941 = -d([2,2,0])$ with the third column in order to return the second column to the degree bounds needed for a Mahler system of type $[2,2,1]$ (see Lemma 5.1(c)). This then gives

$$\mathbf{M}_5 = \begin{bmatrix} 33508\,z^2 - 232744\,z - 324712 & 65690\,z + 87722 & -105364\,z - 122892 \\ 12316\,z - 33802 & 33508\,z^2 - 5906z + 12531 & 2904\,z - 12862 \\ 628930 & -200501 & 33508\,z + 238650 \end{bmatrix}.$$

□

[3]Since $f(0) = [1,9,1] \neq 0$, we have controllable data, and thus we may drop the asterix.

We remark that our use of the integers as a coefficient domain in Example 6.2 is mainly for ease of presentation. A more typical domain would be $\mathbb{Q}[\epsilon]$ where ϵ denotes an indeterminant (for example, ϵ may be a symbolic representation of an allowable error for numeric input).

An asymptotic cost analysis of computing a Mahler system by the algorithm FF-FGnormal is provided in the following theorem. Here we assume following [10] that, for $a, b \in \mathbb{D}$,

$$\begin{aligned} \text{size}(a+b) &= \mathcal{O}(max\{\text{size}(a), \text{size}(b)\}) \\ \text{size}(a \cdot b) &= \mathcal{O}(\text{size}(a) + \text{size}(b)) \\ \text{cost}(a+b) &= \mathcal{O}(1) \\ \text{cost}(a \cdot b) &= \mathcal{O}(\text{size}(a) \cdot \text{size}(b)) \end{aligned}$$

where the function "size" measures the total storage required for its arguments and the function "cost" estimates the number of boolean operations (machine cycles) required to perform the indicated arithmetic. These assumptions are justified for large operands where, for example, the cost of addition is negligible in comparison to the cost of multiplication. Notice that a smaller complexity may be expected if fast multiplication algorithms (e.g., Schönhage–Strassen) can be applied (cf. Knuth [37]).

Theorem 6.3 *Let κ be an upper bound for the size of any element occurring in $\mathbf{C}$ or in $\mathbf{C}^j \cdot \mathbf{F}$, $j \geq 0$, and suppose that only $\mathcal{O}(1)$ entries in a row of $\mathbf{C}$ are different from zero. Then for computing a Mahler system of order K by the algorithm FFFGnormal we have the cost estimate $\mathcal{O}(m \cdot K^4 \cdot \kappa^2)$.*

Proof: Let $0 \leq k \leq K$. We obtain a bound for the size of the $m \times (k+1)$ coefficients of the components of $\mathbf{M}_k$ by using the determinantal representation of Definition 5.2: Applying the Hadamard inequality (8) and taking into account the above assumptions, we get for their size the upper bound $\mathcal{O}(k \cdot \kappa)$. The same size estimate is valid for the $m \cdot (K-k)$ non–trivial components of the residual vectors $\mathbf{R}_k^{(\ell)}$, $\ell = 1, .., m$.

In step k of the algorithm, we have to perform essentially $2m$ operations of the form

$$\mathbf{P}_3 = [a_1 \cdot \mathbf{P}_1 + a_2 \cdot \mathbf{P}_2]/a_3,$$

where $a_j \in \mathbb{D}$, and $\mathbf{P}_j \in \mathbb{D}[z]^m$ having $\mathcal{O}(k)$ nontrivial coefficients. In addition, for computing the residual vectors we again have essentially $2m$ operations of the above form, but now $\mathbf{P}_j$ stands for some vector having $\mathcal{O}(K-k)$ nontrivial components (by assumption on $\mathbf{C}$, the cost of multiplying $(\mathbf{C}_K - c_{k,k} \cdot \mathbf{I}_K)$ with $\mathbf{R}_k^{(\lambda)}$ is negligible). As a consequence, in step k we have $\mathcal{O}(m \cdot K)$ multiplications (and additions) of two elements of $\mathbb{D}$, each being of size bounded by $\mathcal{O}(k \cdot \kappa)$. Summing over $k = 0, .., K-1$ gives the cost estimate as claimed above. □

The cost estimate $\mathcal{O}(m \cdot K^4 \cdot \kappa^2)$ of algorithm FFFGnormal has to be compared with solving (12) by fraction–free Gaussian elimination, with cost given by $\mathcal{O}(K^5 \cdot \kappa^2)$. For the special case of Matrix Padé approximation, we gain a factor m in comparison with the method proposed in [10]. Let us mention already in this context that a modification of FFFGnormal presented in the following section will have the same complexity in case of singularities, whereas the complexity may increase by a factor K for look–ahead methods such as [10].

7 The General Recurrence: Non-perfect Systems

In this section we present an algorithm that avoids non-normal points by travelling around them along a path of "closest para–normal points". We will show that this path of closest para–normal points is separated for each order by at most one unit. The recurrence from Section 6 will then be valid for this problem.

Let $\vec{n} = (\vec{n}^{(1)}, ..., \vec{n}^{(m)})$ be a multi-index. We will construct a sequence of multi-indices $(\vec{n}_k)_{k=0,...,|\vec{n}|}$ with $|\vec{n}_k| = k$ and $\vec{n}_{|\vec{n}|} = \vec{n}$ along an offdiagonal path of indices, namely a particular staircase of the form (15). At the same time we will construct a sequence of multi-indices $(\vec{\nu}_k)_{k=0,...,|\vec{n}|}$ together with the corresponding Mahler systems $\mathbf{M}(\vec{\nu}_k, z)$. These points have the property that $\vec{\nu}_k = \vec{n}_k$ if and only if $\vec{n}_k$ is a normal point. Otherwise, the multi-index $\vec{\nu}_k$ is a k–normal point having a kind of 'minimal distance' to the sequence $(\vec{n}_j)_j$ as specified below (see Theorem 7.3 and the subsequent remarks).

In order to simplify the presentation, we first introduce some properties for $m \times m$ polynomials which will hold for the Mahler systems computed below.

Definition 7.1 ($\vec{n}$–Popov form, Popov–basis)
A $m \times m$ matrix polynomial $\mathbf{M}(z) \in \mathbb{Q}^{m\times m}[z]$ is in $\vec{n}$–Popov form (with row degree $\vec{\nu}$) if there exists a multi-index $\vec{\nu}$ such that $\mathbf{M}(z)$ satisfies the degree constraints

$$z^{-\vec{\nu}} \cdot \mathbf{M}(z) = c \cdot \mathbf{I}_m + \mathcal{O}(z^{-1})_{z\to\infty}, \quad c \in \mathbb{Q} \setminus \{0\}, \tag{17}$$

$$z^{-\vec{n}} \cdot \mathbf{M}(z) \cdot z^{\vec{n}-\vec{\nu}} = \mathbf{T} + \mathcal{O}(z^{-1})_{z\to\infty}, \quad \mathbf{T} \in \mathbb{Q}^{m\times m} \text{ being upper triangular.} \tag{18}$$

If, in addition, the columns of $\mathbf{M}(z)$ have order $\geq \sigma$ with $\sigma \geq \sigma(|\vec{\nu}|)$, then $\mathbf{M}(z)$ will be referred to as a Popov–basis *of type $(\sigma, \vec{n})$.* □

Notice that the matrix $\mathbf{T}$ in (18) is necessarily nonsingular because of (17). Also, by multiplying with an appropriate constant we may suppose that $\mathbf{M}(z)$ has coefficients

in $\mathbb{D}$ (in fact, we will only encounter Mahler systems). Up to a (unique) permutation of columns, we find the classical Popov normal form [36, Subsection 6.7.2, p.481] in the case $c = 1$ and $\vec{n} = \vec{0}$ (or $\vec{n} = [N, N, .., N]$ since (18) is invariant under adding a constant to all components of $\vec{n}$). Here the row degree $\vec{\nu}$ is usually referred to as the vector of *controllability* or *Kronecker indices*. It is known [36, p.484] that any square nonsingular matrix polynomial may be transformed to Popov normal form by multiplication on the right by a unimodular matrix polynomial, and that the resulting polynomial is unique.[4] The introduction of an additional parameter $\vec{n}$ is natural in the context of the approximation problems of Section 2. Also, by an appropriate choice of $\vec{n}$ we may force the matrix $\mathbf{M}(z)$ to be upper triangular, allowing us to include the Hermite normal form in our framework (see, e.g., [36, Subsection 6.7.1, p.476]).

The notion basis will become clear from Theorem 7.3(a) since any solution of order at least σ may be rewritten as a polynomial linear combination of the columns of a Popov–basis of type $(\sigma, \vec{n})$. For solutions of type $(\sigma, \vec{n})$ or, more generally, of type $(\sigma, \vec{n}_k)$ we may even be more precise. In fact, it is easy to see that the set of polynomial vectors of order $\geq \sigma$ forms a submodule over $\mathbb{Q}[z]$ of the module $\mathbb{Q}[z]^m$. Bases of such modules have already successfully computed (not in a fraction–free way) by several authors [3, 5, 8, 9, 18, 19, 21, 39, 16, 51, 52, 53]. Here we may distinguish between two different kinds of algorithms (for a summary, see, e.g., [9]). For the hybrid (or look–ahead) methods in [18, 19, 21, 39, 53] one only uses order bases corresponding to normal or perfect points. In this case additional degree constraints are simple to describe (see, e.g., Corollary 5.4). In contrast, for the single step methods given in [3, 5, 8, 51, 52] only weaker degree constraints are imposed, (for example, there is no longer uniqueness). A rather detailed study of the fine structure of degrees of bases in case of singular Matrix Padé approximation has been given in [16], based on a different computational path and a different normalization of bases. The approach used in this paper of combining order bases with Popov normal forms seems to be conceptionally simpler than that of [16], and easily extends to fraction-free computations.

In the algorithm FFFG (see Table 2) we compute a sequence of para–normal multi-indices $(\vec{\nu}_\sigma)_{\sigma=0,..,K}$ together with the corresponding Mahler systems (up to a sign which may be determined by adapting (16)), using the fraction–free recurrence relation of Theorem 6.1. The efficient computation of the quantities $r^{(\ell)}$ is not specified. It can be implemented as described before Example 6.2. We establish in Theorem 7.2 below the connection to Popov bases. In Theorem 7.3, we show in particular that we have solved the interpolation problem of Definition 2.1.

[4] These properties remain valid for the more general $\vec{n}$–Popov form. A proof of this statement will be given in a future publication where the fraction–free computation of polynomial normal forms is studied. As a consequence, we obtain uniqueness (up to a constant factor) of Popov–bases of a given type. A constructive proof of existence will be given in Theorem 7.2 below. In addition, it follows from Theorem 7.2 that a Popov–basis with row degree $\vec{\nu}$ coincides up to a constant with the (nontrivial) Mahler system $\mathbf{M}(\vec{\nu}, z)$.

Table 2: *The algorithm FFFG*

ALGORITHM FFFG (on off–diagonal staircases)

INPUT: a vector of formal series f, a multi-index $\vec{n}$.

OUTPUT: For $\sigma = 0, 1, 2, \ldots, K$ with $\epsilon_\sigma \in \{-1, 1\}$:
$\vec{\nu}_\sigma$, a closest σ–normal point to $(\vec{n}_k)_{k=0,1,\ldots}$ defined by (15), (19),
Mahler systems $\mathbf{M}_\sigma = \epsilon_\sigma \cdot \mathbf{M}(\vec{\nu}_\sigma, z)$,
multigradients $d_\sigma = \epsilon_\sigma \cdot d^*(\vec{\nu}_\sigma)$,
Basis for set of solutions of type $(\sigma, \vec{n}_k)$, $k \geq 0$:
$\{z^\ell \cdot \mathbf{M}_\sigma^{(\cdot,\mu)} : \ell = 0, 1, ..., \vec{n}_k^{(\mu)} - \vec{\nu}_\sigma^{(\mu)} - 1,\ \mu = 1, ..., m\}$.

INITIALIZATION: $\mathbf{M}_0 \leftarrow \mathbf{I}_m$, $d_0 \leftarrow 1$, $\vec{\nu}_0 \leftarrow \vec{0}$

ITERATIVE STEP: For $\sigma = 0, 1, 2, .., K-1$:
Calculate for $\ell = 1, .., m$:
first term of residuals $r^{(\ell)} \leftarrow c_\sigma(f \cdot \mathbf{M}_\sigma^{(\cdot,\ell)})$
Define set $\Lambda = \Lambda_\sigma = \{\ell \in \{1, .., m\} : r^{(\ell)} \neq 0\}$.

If $\Lambda = \{\}$ **then** $\mathbf{M}_{\sigma+1} \leftarrow \mathbf{M}_\sigma$, $d_{\sigma+1} \leftarrow d_\sigma$, $\vec{\nu}_{\sigma+1} \leftarrow \vec{\nu}_\sigma$
else

Next closest para–normal point: $\vec{\nu}_{\sigma+1} \leftarrow \vec{\nu}_\sigma + \vec{e}_\pi$, where $\pi = \pi_\sigma \in \Lambda$ satisfies
$\pi = \min\{\ell \in \Lambda : \vec{n}^{(\ell)} - \vec{\nu}_\sigma^{(\ell)} = \max_{\mu \in \Lambda}\{\vec{n}^{(\mu)} - \vec{\nu}_\sigma^{(\mu)}\}\}$.

Calculate for $\ell = 1, .., m$, $\ell \neq \pi$:
leading coefficients $p^{(\ell)} \leftarrow \mathit{coefficient}(\mathbf{M}_\sigma^{(\ell,\pi)}, z^{\vec{\nu}_\sigma^{(\ell)} - 1})$.

Increase order for $\ell = 1, .., m$, $\ell \neq \pi$:
$\mathbf{M}_{\sigma+1}^{(\cdot,\ell)} \leftarrow [\mathbf{M}_\sigma^{(\cdot,\ell)} \cdot r^{(\pi)} - \mathbf{M}_\sigma^{(\cdot,\pi)} \cdot r^{(\ell)}]/d_\sigma$
$\mathbf{M}_{\sigma+1}^{(\cdot,\pi)} \leftarrow (z - c_{\sigma,\sigma}) \cdot \mathbf{M}_\sigma^{(\cdot,\pi)}$

Adjust degree constraints:
$\mathbf{M}_{\sigma+1}^{(\cdot,\pi)} \leftarrow [\mathbf{M}_{\sigma+1}^{(\cdot,\pi)} \cdot r^{(\pi)} - \sum_{\ell \neq \pi} \mathbf{M}_{\sigma+1}^{(\cdot,\ell)} \cdot p^{(\ell)}]/d_\sigma$

New multigradient: $d_{\sigma+1} = r^{(\pi)}$
endif

We remark that the algorithm has been implemented in the Maple computer algebra system with the code available from either author.

Theorem 7.2 (Feasibility of the algorithm FFFG)
Algorithm FFFG of Table 2 is well-defined and gives the specified results (see also Theorem 7.3(a)): For any $\sigma \geq 0$, the multi–index $\vec{\nu}_\sigma$ is σ–normal, and $\mathbf{M}_\sigma$ coincides up to a sign with the Mahler system $\mathbf{M}(\vec{\nu}_\sigma, z)$. Furthermore, $\mathbf{M}_\sigma$ is a Popov–basis of type $(\sigma, \vec{n})$ with row–degree $\vec{\nu}_\sigma$. Finally, with the assumptions of Theorem 6.3, computing a Mahler system of order K has a cost of $\mathcal{O}(m \cdot K^4 \cdot \kappa^2)$.

Proof: The first sentence of the assertion follows by recurrence on σ from Theorem 6.1, where in contrast to algorithm FFFGnormal we also apply Theorem 6.1(a) in order to include the case $\Lambda = \{\}$. Let us verify (again by induction on σ) the link to Popov bases. The assertion is trivially true for $\sigma = 0$ since $\mathbf{M}_0 = \mathbf{I}_m$. Suppose therefore that the statement holds for $\sigma \geq 0$, and let $\Lambda \neq \{\}$ (the case $\Lambda = \{\}$ is trivial). In the sequel of the proof we write $\vec{\nu} = \vec{\nu}_\sigma$ and recall that $\vec{\nu}_{\sigma+1} = \vec{\nu} + \vec{e}_\pi$. We know already that $\mathbf{M}_{\sigma+1}$ is an Mahler system of type $\vec{\nu} + \vec{e}_\pi$, implying that the conditions on the order of the columns as well as the degree constraints (17) hold. To verify (18), notice that the recurrence of Theorem 6.1(c) may be rewritten in the form

$$\mathbf{M}_{\sigma+1} = \mathbf{M}_\sigma \cdot \mathbf{P}_1 \cdot \mathbf{P}_2 \cdot d,$$

with some $d \in \mathbb{Q}$, and

$$\mathbf{P}_1 = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ a_1 & \cdots & a_{\pi-1} & z - a_\pi & a_{\pi+1} & \cdots & a_m \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}, \mathbf{P}_2 = \begin{bmatrix} 1 & & & b_1 & & & \\ & \ddots & & \vdots & & & \\ & & 1 & b_{\pi-1} & & & \\ & & & 1 & & & \\ & & & b_{\pi+1} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & b_m & & & 1 \end{bmatrix},$$

with some $a_j, b_j \in \mathbb{Q}$, where we have not displayed the zero entries. By hypothesis we know that the limit for $z \to \infty$ of $z^{-\vec{n}} \cdot \mathbf{M}_\sigma \cdot z^{\vec{n}-\vec{\nu}}$ is upper triangular. Also, the limit of $z^{-\vec{n}+\vec{\nu}} \cdot \mathbf{P}_1 \cdot z^{\vec{n}-\vec{\nu}-\vec{e}_\pi}$ is upper triangular since, by construction of π, for $\ell = 1, ..., m$ we have either $a_\ell = 0$ or $\vec{n}^{(\pi)} - \vec{\nu}^{(\pi)} \geq \vec{n}^{(\ell)} - \vec{\nu}^{(\ell)}$, with equality only if $\ell \geq \pi$. In addition, $b_\ell \neq 0$ implies that $deg\, \mathbf{M}_\sigma^{(\ell,\pi)} = \vec{\nu}^{(\ell)} - 1$. Comparing with (18) for $\mathbf{M}_\sigma$, we get for $\ell = 1, ..., m$ either $b_\ell = 0$ or $\vec{n}^{(\pi)} - \vec{\nu}^{(\pi)} - 1 \leq \vec{n}^{(\ell)} - \vec{\nu}^{(\ell)}$, with equality only if $\ell < \pi$. Thus the limit of $z^{-\vec{n}+\vec{\nu}+\vec{e}_\pi} \cdot \mathbf{P}_2 \cdot z^{\vec{n}-\vec{\nu}-\vec{e}_\pi}$ is upper triangular, and we have established relation (18) for $\mathbf{M}_{\sigma+1}$. Finally, for a cost analysis we may fully apply the reasoning of the proof of Theorem 6.3 □

In order to describe further properties of the output of algorithm FFFG, let us specify the offdiagonal path mentioned in the introduction of this section. The sequence

$(\vec{n}_k)_{k=0,1,2,...}$ is constructed at each step by increasing the index that has the furthest to go to reach $\vec{n}$, with ties broken by index order. That is, given a $\vec{n}_k$ we determine $\vec{n}_{k+1}$ by increasing the λ-th component by one where λ is chosen as $\vec{n}^{(\lambda)} - \vec{n}_k^{(\lambda)} = \max_\mu\{\vec{n}^{(\mu)} - \vec{n}_k^{(\mu)}\}$. If there is more than one choice of λ then λ is the minimum index satisfying the maximality condition. In other words, we use the construction (15), where in each step $\lambda = \lambda_k$ satisfies

$$\lambda := \min\{\ell \in \{1,..,m\} : \vec{n}^{(\ell)} - \vec{n}_k^{(\ell)} = \max_{\mu\in\{1,..,m\}}\{\vec{n}^{(\mu)} - \vec{n}_k^{(\mu)}\}\}. \tag{19}$$

Thus, for example if $\vec{n} = [1,3,3]$ then the sequence of 8 vectors are $[0,0,0]$, $[0,1,0]$, $[0,1,1]$, $[0,2,1]$, $[0,2,2]$, $[1,2,2]$, $[1,3,2]$ and $[1,3,3]$. In particular, notice that $\vec{n}_{|\vec{n}|} = [1,3,3]$, the multi-index of our original problem. The choice (19) of our particular staircase can also be understood as an elimination strategy in an extrapolation process with respect to an asymptotic scale of comparison. In fact, because of some numerical and theoretical results, this ordering was also preferred for GREP [28].

Theorem 7.3 (Properties of the algorithm FFFG)
(a) *For all $k, \sigma \geq 0$, the set of solutions of type $(\sigma, \vec{n}_k)$ (and thus the kernel of the matrix $\mathbf{K}(\vec{n}_k, \sigma)$) is spanned by*

$$z^j \cdot \mathbf{M}_\sigma^{(\cdot,\mu)}, \quad j = 0, 1, ..., \vec{n}_k^{(\mu)} - \vec{\nu}_\sigma^{(\mu)} - 1, \quad \mu = 1, ..., m.$$

(b) *For all $k, \sigma \geq 0$ we have*[5] rank $\mathbf{K}(\vec{n}_k, \sigma) = |\min(\vec{\nu}_\sigma, \vec{n}_k)|$.
(c) *A multi–index $\vec{\nu}$ verifying* rank $\mathbf{K}(\vec{n}_k, \sigma) = |\min(\vec{\nu}, \vec{n}_k)|$ *for $k \geq 0$ necessarily coincides with $\vec{\nu}_\sigma$.*
(d) *The multi-index $\vec{n}_k$ is σ–normal iff $\vec{n}_k = \vec{\nu}_\sigma$. In particular, $\vec{n}_k$ is normal iff $\vec{n}_k = \vec{\nu}_k$.*

Proof: For a proof of (a),(b), let us first mention that the $K := \max(\vec{0}, \vec{n}_k - \vec{\nu}_\sigma) = |\vec{n}_k| - |\min(\vec{\nu}_\sigma, \vec{n}_k)|$ polynomial vectors given in the assertion are linearly independent over $\mathbb{Q}$ by Lemma 5.3(e) since $\mathbf{M}_\sigma$ essentially is a Mahler system. Let us show that they are all solutions of type $(\sigma, \vec{n}_k)$. From Theorem 7.2 we know that the order is correct. Furthermore, from (18) we have the degree constraints

$$deg\, M_\sigma^{(\ell,\mu)} \leq \vec{n}^{(\ell)} - \vec{n}^{(\mu)} + \vec{\nu}_\sigma^{(\mu)} - \eta_{\ell,\mu}, \quad \ell, \mu = 1, .., m,$$

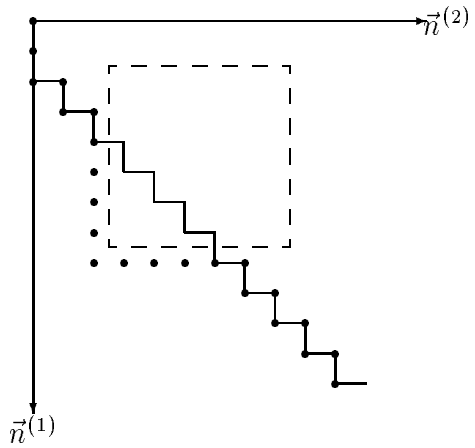
with $\eta_{\ell,\mu} = 1$ if $\ell > \mu$ and $\eta_{\ell,\mu} = 0$ otherwise. Notice also that our offdiagonal staircase verifies

$$\vec{n}_k^{(\mu)} > 0 \quad \Longrightarrow \quad \vec{n}^{(\mu)} - \vec{n}_k^{(\mu)} \geq \vec{n}^{(\ell)} - \vec{n}_k^{(\ell)} - \eta_{\ell,\mu}, \quad \ell, \mu = 1, ..., m.$$

Hence in the case $\vec{n}_k^{(\mu)} - \vec{\nu}_\sigma^{(\mu)} > 0$ (and thus $\vec{n}_k^{(\mu)} > 0$) we get $deg\, M_\sigma^{(\ell,\mu)} \leq \vec{n}_k^{(\ell)} - \vec{n}_k^{(\mu)} + \vec{\nu}_\sigma^{(\mu)}$, as required to show that the polynomial vectors of (a) are solutions of type $(\sigma, \vec{n}_k)$. Consequently, we have found K linearly independent elements of the kernel of $\mathbf{K}(\vec{n}_k, \sigma)$, showing that *rank* $\mathbf{K}(\vec{n}_k, \sigma) \leq |\min(\vec{\nu}_\sigma, \vec{n}_k)|$. On the other hand, by

[5] In what follows, the operations max, min for integer vectors are defined on a component basis.

Table 3: *An example of singular Padé approximation. We have drawn the corresponding C–table of Bigradients, here the dashed square indicates a singular block of zero–entries. By a straight line we denote the offdiagonal path induced by $\vec{n} = (7,6)$, with the dots characterizing the corresponding closest para–normal points.*



Lemma 4.3 and the para–normality of $\vec{\nu}_\sigma$, a nonsingular submatrix of $\mathbf{K}(\vec{n}_k, \sigma)$ is given by $\mathbf{K}^*(\min(\vec{\nu}_\sigma, \vec{n}_k), |\min(\vec{\nu}_\sigma, \vec{n}_k)|)$, implying the assertions (a),(b).

In order to show (c), notice that by assumption and part (b) we have $|\min(\vec{\nu}_\sigma, \vec{n}_k)| = |\min(\vec{\nu}, \vec{n}_k)|$ for all $k \geq 0$. If now $\vec{\nu} \neq \vec{\nu}_\sigma$, say, $\vec{\nu}^{(\ell)} < \vec{\nu}_\sigma^{(\ell)}$, then we may find a k such that $\vec{n}_k^{(\ell)} = \vec{\nu}^{(\ell)}$, and $\vec{n}_{k+1}^{(\mu)} = \vec{n}_k^{(\mu)} + \delta_{\ell,\mu}$, a contradiction.

Finally, for part (d) it is sufficient to prove the first sentence since $|\vec{n}_k| = k$. The σ–normality of $\vec{\nu}_\sigma$ has been already established in Theorem 7.2, and the final implication is a consequence of Lemma 4.3 and part (c). □

Notice that, for any k, and for any σ–normal multi-index $\vec{\nu}$, the matrix $\mathbf{K}(\vec{a}, \sigma)$, $\vec{a} := \min\{\vec{\nu}, \vec{n}_k\}$, is a submatrix both of $\mathbf{K}(\vec{n}_k, \sigma)$, and of $\mathbf{K}(\vec{\nu}, \sigma)$. Moreover, by Lemma 4.3, the latter matrix has maximal column rank. Therefore the Krylov matrix $\mathbf{K}(\vec{a}, \sigma)$ has full column rank, and, by Theorem 7.3(b),

$$|\max\{\vec{0}, \vec{n}_k - \vec{\nu}\}| = |\vec{n}_k| - \mathit{rank}\ \mathbf{K}(\vec{a}, \sigma) \geq |\vec{n}_k| - \mathit{rank}\ \mathbf{K}(\vec{n}_k, \sigma) = |\max\{\vec{0}, \vec{n}_k - \vec{\nu}_\sigma\}|. \quad (20)$$

Thus one may consider $\vec{\nu}_\sigma$ as the *closest σ–normal point* to the sequence $(\vec{n}_k)_k$, and in addition such a multi–index is unique according to Theorem 7.3(c). In order to illustrate this statement, we have drawn in Table 3 in the classical C–table (i.e., Padé approximation, $m = 2$) an off–diagonal path together with the path of closest para–normal points. We also remark that the classic block structure of the Padé table is easily shown using Theorem 7.3(a) (cf. [9, Example 4.2]).

We may now establish the equivalent characterization of para–normal points as claimed in Corollary 5.4

Proof of Corollary 5.4: If $\vec{n}$ is σ–normal, then $\mathbf{M}(z) = \mathbf{M}(\vec{n}, z)$ has the required properties by construction. Let therefore $\mathbf{M}(z)$ be given as described above. We have shown in Theorem 7.2 that $\vec{\nu} := \vec{\nu}_\sigma$ is σ–normal, and hence $\sigma(|\vec{\nu}| - 1) < \sigma \leq \sigma(|\vec{\nu}|)$, implying that $|\vec{\nu}| \geq |\vec{n}|$. On the other hand, the columns of $\mathbf{M}(z)$ are all solutions of type $(\sigma, \vec{n})$. Thus from Theorem 7.3(a) we know that there exist a matrix polynomial $\mathbf{P}(z)$ such that $\mathbf{M}(z) = \mathbf{M}_\sigma \cdot \mathbf{P}(z)$. Taking into account the degree assumption on $\mathbf{M}(z)$ and (18), we may conclude that the limit of $z^{\vec{\nu}-\vec{n}} \cdot \mathbf{P}(z)$ for $z \to \infty$ exists. Hence the components of $\vec{n} - \vec{\nu}$ have to be nonnegative, which together with $|\vec{\nu}| \geq |\vec{n}|$ implies that $\vec{\nu} = \vec{n}$. Consequently, $\vec{n}$ is σ–normal, and the representation of Corollary 5.4 follows from Lemma 5.3(c). □

Besides solving the approximation problems of Section 2, we mention one further application for algorithm FFFG

Example 7.4 (Fraction-free Hankel Matrix Solver)
Suppose that we want to solve a system of linear equations

$$\mathbf{H} \cdot x = b, \quad \mathbf{H} = [h_{i+j}]_{i,j=0..n-1}, \quad b = [b_j]_{j=n-1,\ldots,2n-2},$$

with a Hankel matrix of coefficients. If $h_j, b_j \in \mathbb{D}$, we may apply FFFG in two different ways to obtain the Cramer solution $x^* := x \cdot det\,\mathbf{H} \in \mathbb{D}^n$: First, as mentioned already in the context of equation (5), we may consider a (homogenous) Hermite Padé approximation problem with $m = 3$, $f_1 = -1$, $f_2(z) = \sum h_j z^j$, $f_3(z) = -\sum b_j z^j$. It is easily shown that the resulting Sylvester matrix $\mathbf{K}(\vec{n}, 2n-1)$, $\vec{n} := [n-1, n, 0]$, is upper block triangular, with the left upper block being equal to the identity of order $n-1$, and the right lower block being equal to $\mathbf{H}$ up to a permutation of the columns. Hence $det\,\mathbf{H} = \pm d(\vec{n})$ and $\mathbf{H}$ is nonsingular iff $\vec{n}$ is normal. In this case, $\mathbf{M}^{(3,3)}(\vec{n}, z) = \epsilon \cdot det\,\mathbf{H}$, $\epsilon \in \{\pm 1\}$, and thus the coefficient vector of $\mathbf{M}^{(2,3)}(\vec{n}, z)$ is ϵ times the Cramer solution x^* of our Hankel system.

If one wants to solve the above system for multiple right hand sides, it may be more interesting to get explicitly the adjoint $det\,\mathbf{H} \cdot \mathbf{H}^{-1} \in \mathbb{D}^{n \times n}$. Again this can be done by FFFG, using a well–known inversion formula for Hankel matrices: we compute the Mahler system $\mathbf{M}([n-1, n], z)$ corresponding to the Padé approximation problem $f = [f_1, f_2]$, and denote by $[q_j]_{j=0,\ldots,n-1}$, and $[v_j]_{j=0,\ldots,n}$, respectively, the coefficient vectors of $\mathbf{M}^{(2,1)}([n-1, n], z)$, and of $\mathbf{M}^{(2,2)}([n-1, n], z)$, $q_n := 0$. Then $v_n = \pm det\,\mathbf{H}$, and from [35, Section 1] we obtain (up to a sign) the adjoint of $\mathbf{H}$ by

$$v_n \cdot \mathbf{H}^{-1} = \frac{1}{\mathrm{v_n}} \cdot \begin{bmatrix} v_n & & \\ \vdots & \ddots & \\ v_1 & \cdots & v_n \end{bmatrix} \begin{bmatrix} q_{n-1} & \cdots & q_0 \\ \vdots & . & \\ q_0 & & \end{bmatrix} - \frac{1}{\mathrm{v_n}} \cdot \begin{bmatrix} q_n & & \\ \vdots & \ddots & \\ q_1 & \cdots & q_n \end{bmatrix} \begin{bmatrix} v_{n-1} & \cdots & v_0 \\ \vdots & . & \\ v_0 & & \end{bmatrix}$$

Thus our algorithm gives a fraction-free method of solving systems of equations having Hankel coefficient matrices. By reversing the order of columns one can state a similar result for Toeplitz systems. Note that in this case we have a Hankel, rather than Toeplitz solver, since the algorithm solves all subproblems for nonsingular matrices along the principal diagonal of a Hankel matrix (principal anti-diagonal of a Toeplitz matrix). The complexity of this solver is $\mathcal{O}(n^4 \cdot \kappa^2)$ which is faster than $\mathcal{O}(n^5 \cdot \kappa^2)$ required for fraction–free Gaussian elimination. One can also use our algorithm for fast fraction-free solving of linear systems having coefficient matrices that are block Hankel or block Hankel-like [39]. □

8 Fraction-free matrix GCD computations

Given two matrix polynomials A, B having s rows, with elements in $\mathbb{D}[z]$, the aim of this section is to show that the algorithm FFFG of Section 7 enables us to compute a greatest common left divisor (GCLD) of A, B in a fraction–free way. Here it is convenient to combine A, B in a larger matrix $G = [A, B] \in \mathbb{D}[z]^{s \times m}$, where we suppose[6] that the rows of G are linearly independent over $\mathbb{Q}[z]$. We recall the well–known fact (see, e.g., [36, Lemma 6.3-3, p.377]) that from a decomposition

$$G \cdot U = [A, B] \cdot U = [R, 0], \quad R \in \mathbb{Q}[z]^{s \times s}, \quad U \in \mathbb{Q}[z]^{m \times m}, \tag{21}$$

with U being unimodular (i.e., $detU \in \mathbb{Q} \setminus \{0\}$) we may read off the solution of the matrix GCD problem: the matrix R is a GCLD (over $\mathbb{Q}[z]$) of A, B, and it is unique up to multiplication on the right by a unimodular matrix (in particular, the degree of its determinant is unique). Note that, by multiplying with a suitable element from $\mathbb{D}$, the matrices R, U of (21) may be chosen to have elements only from $\mathbb{D}[z]$. The algorithm FFFG will not only provide $R \in \mathbb{D}[z]^{s \times s}$ but also the cofactor matrix U.

The link to the interpolation problems of Section 2 is given by reversing coefficients, i.e., z is replaced by $1/z$. We are then left with a vector Hermite–Padé approximation problem, with the corresponding system of functions $f \in \mathbb{D}[z]^{s \times m}$ being polynomial. However, the corresponding $(\mathbf{C}, \mathbf{F})$ is in general not controllable. Some results for the recursive solution of such a problem have been mentioned (without complete proofs) in [8, Section 4] by exploiting the connections to power Hermite–Padé approximants.

In order to describe the complexity of our approach, we will make use of a result of Kung, Kailath and Morf [38], [11, Theorem 1] on the rank of certain block Sylvester matrices. Here we require some definitions from the theory of matrix polynomials: The

[6] This restriction is natural since otherwise we may have GCDs with arbitrarily high degree [11], [36, p.376ff].

degree of a (rectangular) matrix polynomial C is the smallest integer N allowing a representation of the form $C(z) = C_0 + C_1 z + ... + C_N z^N$. The *McMillan degree* of C is the maximum of the degrees of the determinant of a maximal square submatrix of C (see, e.g., [36, 11, 51, 52]). We also need the concept of *minimal indices* [36, Section 6.5.4] which are closely related to the controllability and Kronecker indices mentioned previously. The solutions $h \in \mathbb{Q}[z]^m$ of the equation $G \cdot h = 0$ form a submodule $\mathcal{M}$ of $\mathbb{Q}[z]^m$ of dimension $m-s$. We may find a basis of $\mathcal{M}$ given by the columns of $H = [h_1, ..., h_{m-s}] \in \mathbb{Q}[z]^{m \times (m-s)}$ such that H is column–reduced and irreducible [36, Theorem 6.5-10, p.458]. Denoting by $\vec{\alpha}^{(j)}$ the degree of h_j, $j = 1, .., m-s$, and $\vec{\alpha} = (\vec{\alpha}^{(1)}, ..., \vec{\alpha}^{(m-s)})$, it is known that $\vec{\alpha}$ is unique (up to a permutation) [36, Lemma 6.3-14], and that $|\vec{\alpha}|$ equals the McMillan degree of G minus the degree of the determinant of an GLCD.

We state the main result of this section in the following

Theorem 8.1 (GCLD via FFFG)
Let $G = [A, B] \in \mathbb{D}[z]^{s \times m}$ with degree N and McMillan *degree $N^{\#}$. In addition, let $\vec{\alpha}$ be the vector of minimal indices of G, with its largest component*[7] *denoted by N^*. If we apply algorithm FFFG to the data $f(z) = z^N \cdot G(1/z)$, $\vec{n} = (N, N, ..., N)$, $c_{j,k} = \delta_{j-s,k}$, with stopping criterion: $\sigma = \sigma^*$ such that $f \cdot \mathbf{M}_\sigma$ is* reduced, *i.e., $f \cdot \mathbf{M}_\sigma$ contains only s columns different from zero, then*
(a) For $\sigma \geq \sigma^*$, the matrix $U_\sigma(z)$, a column permutation of $\mathbf{M}_\sigma(1/z) \cdot z^{\vec{\nu}_\sigma}$, is unimodular and verifies (21). Thus we have solved the extended GLCD problem.
(b) We have that $\sigma^* \leq \sigma' := s \cdot (N + N^* + 1)$. For N^* we have the worst case[8] *estimate $N^* \leq N^{\#} \leq s \cdot N$.*
(c) Suppose the coefficients occurring in G are all bounded in size by the constant κ. Then computing the GLCD by the algorithm FFFG has a worst case complexity of $\mathcal{O}(m\ (\sigma')^4\ \kappa^2)$.

Proof: Denote by I the set of indices of the columns in $f \cdot M_{\sigma^*}$ which are different from zero. Note that I contains s elements by assumption on G and σ^*. Let $j \in \{1, .., m\} \setminus I$, and $\sigma \geq \sigma^*$. Since $ord(f \cdot \mathbf{M}_{\sigma^*}^{(\cdot,j)}) = \infty$, one easily verifies by induction that the j-th column of $\mathbf{M}_\sigma$ coincides with that of $\mathbf{M}_{\sigma^*}$ (up to some constant). In particular, $f \cdot \mathbf{M}_\sigma$ is reduced. For the assertion of part (a) it remains to show that $U(z) := \mathbf{M}_\sigma(1/z) \cdot z^{\vec{\nu}_\sigma}$ is a unimodular matrix polynomial. In fact, U is a matrix polynomial according to (18), and one easily verifies that $det\,\mathbf{M}_\sigma = d \cdot z^{|\vec{\nu}_\sigma|}$ with $d \in \mathbb{D} \setminus \{0\}$. Therefore $det U \in \mathbb{D}$, as claimed in part (a).

The set Λ appearing in the FFFG algorithm is a subset of I in any step where

[7] If (without loss of generality) the McMillan degree of G is attained for $det\,A$, then N^* is the minimal degree of a matrix polynomial $[C^T, D^T]$ allowing a representation $A^{-1} \cdot B = C \cdot D^{-1}$.

[8] As seen from the proof, it is more likely that N^* has the same magnitude as $N^{\#}/(m-s)$. In this case, σ' is at most of order $(N+1) \cdot s \cdot m/(m-s)$.

$\sigma \geq \sigma^*$. Therefore the components of $\vec{\nu}_\sigma$ with indices not in I remain invariant for $\sigma \geq \sigma^*$. For the remainder of the proof it will be convenient to reorder the columns of G (and thus simultaneously the rows and columns of $\mathbf{M}_\sigma$) such that $I = \{1, 2, ..., s\}$. Thus $U_\sigma(z) = \mathbf{M}_\sigma(1/z) \cdot z^{\vec{\nu}_\sigma}$ and $\vec{\nu}_\sigma = (\vec{c}_\sigma, \vec{a})$ for $\sigma \geq \sigma^*$, with some multi–index $\vec{a}$ having $m - s$ components.

In [11, Theorem 1] (see also [38]), the authors discuss the rank of transposed block Sylvester matrices which are given by $S_k := \mathbf{K}([k, ..., k], s \cdot (k + N))^T$ using our notation. It is shown that

$$rank\ S_k = |\min([\underbrace{k, .., k}_{m}], [\underbrace{k, .., k}_{s}, \vec{\alpha}])|, \quad k \geq 0,$$

where $\vec{\alpha}$ is the vector of minimal indices of G. Notice that $\mathbf{K}([k, ..., k], s \cdot (k + N))$ has a rhombus block structure, and that $\mathbf{K}([k, ..., k], \sigma)$ is obtained from $\mathbf{K}([k, ..., k], s \cdot (k+N))$ for $\sigma \geq s \cdot (k+N)$ by bordering $\sigma - s \cdot (k+N)$ zero rows. Consequently, with $\sigma = s \cdot (N+\ell)$, we get for $\ell = 0, 1, 2, ...$ and for $k = 0, 1, ..., \ell$

$$rank\ \mathbf{K}([k, ..., k], \sigma) = |\min([\underbrace{k, .., k}_{m}], [\underbrace{k, .., k}_{s}, \vec{\alpha}])| = |\min([\underbrace{k, .., k}_{m}], \vec{\nu}_\sigma)|,$$

the final equality following from Theorem 7.3(b). We may conclude that the one partition of $\vec{\nu}_{\sigma'}$, namely $\vec{a}$, coincides up to a permutation with the vector $\vec{\alpha}$ of minimal indices, and that the other partition $\vec{c}_{\sigma'}$ contains only components strictly larger than N^*. Consider now $P(z) := H(1/z) \cdot z^{\vec{\alpha}}$, with $H \in \mathbb{Q}[z]^{m \times (m-s)}$ constituting a minimal basis as described before Theorem 8.1. Then $P \in \mathbb{Q}[z]^{m \times (m-s)}$, with its jth column having the degree $\vec{\alpha}^{(j)}$, and $f \cdot P = 0$. By Theorem 7.3(a), the columns of P may be represented as a polynomial linear combination of the columns of $\mathbf{M}_{\sigma'}$, that is, there exists a $Q \in \mathbb{Q}[z]^{m \times (m-s)}$ such that $P = \mathbf{M}_{\sigma'} \cdot Q$, and $\vec{z}^{\vec{\nu}_{\sigma'}} \cdot Q \cdot z^{-\vec{\alpha}}$ has a finite limit for $z \to \infty$. According to the special form of $\vec{\nu}_{\sigma'}$ we may conclude that the first s rows of Q vanish. Moreover, denoting by Q^* the (square) submatrix obtained from the last $m - s$ rows of Q, we know that $\vec{z}^{\vec{a}} \cdot Q^* \cdot z^{-\vec{\alpha}}$ has a finite limit. In addition, as with P, the columns of Q^* are also linearly independent over $\mathbb{Q}[z]$, and $|\vec{a}| = |\vec{\alpha}|$. Thus Q^* is unimodular, showing that $f \cdot \mathbf{M}_{\sigma'}$ is reduced, and hence $\sigma' \geq \sigma^*$. For a proof of part (b), it remains to establish the (rough) bound for N^*. Notice that $N^* \leq |\vec{\alpha}|$, with the latter quantity being bounded above by $N^{\#}$, the McMillan degree of G (see the remark before Theorem 8.1). The final estimate $N^{\#} \leq s \cdot N$ of part (b) is trivial. Finally, part (c) is a consequence of Theorem 7.2. □

Example 8.2
Let

$$A^*(z) := \begin{bmatrix} 1 - 3\,z & 4\,z \\ 1 & -2 \end{bmatrix}, B^*(z) := \begin{bmatrix} 1 + 2\,z & -4\,z \\ z^2 & 3 \end{bmatrix}, C(z) := \begin{bmatrix} 1 + 3\,z & -3\,z \\ z^2 & -z + z^2 \end{bmatrix}.$$

We will compute the GCLD of the two matrix polynomials $A = C \cdot A^*$ and $B = C \cdot B^*$ using the method described in Theorem 8.1. The matrix polynomials A^* and B^* are shown to be left coprime, and so GCLD's of A and B are obtained by multiplying C on

the right by some unimodular matrix. Here the combined matrix $G(z) = [A(z), B(z)]$ is given by

$$\begin{bmatrix} -9\,z^2 - 3\,z + 1 & 12\,z^2 + 10\,z & -3\,z^3 + 6\,z^2 + 5\,z + 1 & -12\,z^2 - 13\,z \\ -3\,z^3 + 2\,z^2 - z & 4\,z^3 + 2\,z - 2\,z^2 & z^4 + z^3 + z^2 & -4\,z^3 - 3\,z + 3\,z^2 \end{bmatrix},$$

with $m = 4$, $s = 2$ and $N = 4$. We compute that $N^{\#} = 6 < s \cdot N$, while the vector of minimal indices is given by $\vec{\alpha} = (1, 2)$ (see below), and thus $N^* = 2$. Notice that $f(z) = z^N \cdot G(1/z)$ leads to a vector Hermite Padé approximation problem where the data is not controllable (in fact, the first row of f is divisible by z^2). From Theorem 8.1 we know that algorithm FFFG gives us a reduced basis (and thus a GCLD) at iteration σ^*, with $\sigma^* \leq \sigma' = 14$.

Using FFFG we find that $\sigma^* = 11$, and $\vec{\nu}_{\sigma^*} = [3, 3, 2, 1]$ and hence we have computed $|\vec{\nu}_{11}| = 9$ different Mahler systems. It is quite instructive to have a look at the sequence of closest para–normal points $(\vec{\nu}_\sigma)_{0 \leq \sigma \leq \sigma^*}$ which are given by

$$[0,0,0,0],\ [0,0,0,0],\ [0,0,1,0],\ [0,0,1,0],\ [1,0,1,0],\ [1,0,2,0],$$
$$[1,1,2,0],\ [1,1,2,1],\ [2,1,2,1],\ [2,2,2,1],\ [3,2,2,1],\ [3,3,2,1].$$

Clearly, this staircase differs significantly from the off–diagonal staircase induced by $\vec{n} = [4, 4, 4, 4]$, that is, the "ideal" staircase contains only 2 para–normal points, and $[0, 0, 0, 0]$ is the only (trivially) normal point (the linear functionals c_0 and c_2 have been rejected). This illustrates why the reliable version of FFFG as presented in Section 7 is in fact needed.

We note some interesting points about the output of FFFG. By reversing coefficients in $f \cdot \mathbf{M}_{11}$ and by eliminating the last two zero columns, we get the GCLD of A and B as the answer

$$C^*(z) := \begin{bmatrix} -20736 & -124416\,z \\ -41472\,z^2 + 20736\,z & 41472\,z^2 - 41472\,z \end{bmatrix} = -20736 \cdot C(z) \cdot \begin{bmatrix} 1 & 0 \\ 1 & -2 \end{bmatrix},$$

with the factor on the right being unimodular. We observe in this example that the coefficients of the GCLD computed by FFFG still have a common factor $d_{11} = -20736$. However, the prediction of such common factors (which also occur for Cramer solutions in other contexts) seems to be quite a difficult problem to solve. Also, notice that during our intermediate computations we have already factored out $\prod_{j=0}^{10} d_j$, a quantity which is of much bigger size than d_{11}. Finally, we observe that by partitioning

$$U(z) = \mathbf{M}_{11}(1/z) \cdot z^{\vec{\nu}_{11}} = \begin{bmatrix} U_1 & U_2 \\ U_3 & U_4 \end{bmatrix}$$

with blocks of size 2×2 we have found the cofactors in the diophantine equation $A \cdot U_1 + B \cdot U_2 = C^*$. Furthermore, $U_4 \cdot U_3^{-1}$ is the (irreducible) right coprime matrix fraction description of the rational function $B^{-1}A$.

For presentation purposes our example uses coefficients from the integers. A similar example could easily be constructed where the problem has parameters, for example having coefficients from the domain $\mathbb{Q}[\epsilon]$, with ϵ an unknown. □

The significance of Mahler systems for the scalar GCD problem has been discussed in some detail in [10, Section 6]. Here A, B are scalar polynomials, i.e., $m = 2$, $s = 1$, $N = \max(deg\, A, deg\, B) = N^{\#}$, and $N - N^*$ is the degree of the GCD C of A and B. The dimension of the largest Sylvester matrix encountered in FFFG will be $N + N^*$, which may be larger than $deg\, A + deg\, B - deg\, C$, the dimension of the well-known critical Trudi submatrix. In fact, for a more efficient implementation one may choose instead of $\vec{n} = [N, ..., N]$ the "smallest" multi–index $\vec{n}$ such that $f(z) := G(1/z) \cdot z^{\vec{n}}$ is polynomial. Here the corresponding unimodular matrix is obtained by $z^{\vec{n}} \cdot \mathbf{M}_\sigma(1/z) \cdot z^{-\vec{n}+\vec{\nu}_\sigma}$.

9 Conclusions

In this paper we have presented algorithms for the computation of matrix rational interpolants and one-sided matrix greatest common divisors. The algorithms are fraction-free and designed to work in exact arithmetic domains where coefficient growth is a primary concern. The algorithms require no restrictions on input and are at least an order of magnitude faster than existing methods that compute solutions to the general problem. When specialized to cases such as Padé and matrix Padé approximation and scalar greatest common divisor computation, our approach is at least as efficient as existing fast fraction-free algorithms that work for these particular cases [10, 15, 20, 24].

Our method finds a basis for the $\mathbb{Q}[z]$–submodule of polynomial vectors of a given order, by recursively computing all bases of lower order. As such we find all possible solutions to the above interpolation problems. The methods also illustrate the advantages of considering the "closest normal points" of a given off–diagonal staircase of multi–indices which may contain non–normal points. The approach taken in this paper differs from the method proposed in [10], which computes matrix Padé approximation by also using Mahler systems as its fundamental computation tool, but only at normal points. Problems corresponding to non–normal points are "jumped" using fraction-free Gaussian elimination.[9] As a result, in cases where there are significant sized jumps their algorithm is potentially an order of magnitude less efficient than the one presented in this paper.[10]

[9]The method of "jumping" over singularities by some look–ahead strategy has been shown to be very useful in a numerical setting, see [6, 19, 22, 53]. Also, as shown in [10], there is a nice interpretation of such jumps in terms of modified Schur complements.

[10]Jumps of larger size are quite typical for Matrix GCD computations, see Example 8.2.

In the case of computing a scalar GCD, we do not use pseudo–divisions in order to jump over problems associated to multi–indices being not (para–)normal. This is in contrast to classical fraction–free methods for solving such problems [29]. In fact, we do not believe that our algorithm can be easily converted to recover the subresultant algorithm. Instead it is probably the case that one would have to choose bases different from Mahler systems ("comonic" instead of "monic" bases in the terminology of [9]), leading to some fraction-free variant of the algorithm of [16]. However, notice that, for large jumps, the size of the intermediate quantities in the subresultant algorithm [15, 24] (as well as in the algorithms of [10, 20]) may become significant. Our method, using closest normal points, does not have this drawback.

For some applications, it is of interest to follow computational paths different than the off-diagonal paths used in this paper. For example, it is of interest to obtain a Toeplitz instead of a Hankel solver. If this path consists of normal points then one may apply the fraction–free algorithm of Section 6. However, we are interested in giving a version that allows us to drop any regularity assumptions. Here, it might be possible to adapt the method of [5] to fraction–free arithmetic (or, alternatively, the methods [9, 52]). In addition, in some applications such as Padé–Chebyshev approximation or state–space realizations in the theory of linear systems, one is interested in the case where the matrices $\mathbf{C}$ are lower Hessenberg instead of lower triangular. The corresponding special multiplication rule has the drawback that one decreases by one the order while multiplying by z. It is possible to adapt the algorithm FFFGnormal, but a generalization to singular cases is still an open problem.

As mentioned towards the end of Section 2, the computation of matrix rational interpolants are related to the computation of both Popov and Hermite normal forms for matrices of polynomials. We plan to develop efficient fraction-free algorithms for these important computations, by combining our algorithm FFFG with methods presented in [55]. Similarly it is of interest to see if our methods can be extended to Ore domains as done by Li [43] in the case of greatest common divisor computations of differential and difference operators.

Fraction-free algorithms are often important for theoretic reasons since they form the basis for generating exact algorithms based on modular reduction. We plan to investigate such algorithms for computing rational interpolants and matrix greatest common divisors. That these methods ultimately provide improved practical algorithms has been noted by Li [43] in the case of computing greatest common divisors of differential operators.

References

[1] G.A. Baker & P.R. Graves-Morris, *Padé Approximants* , second edition, Cambridge Univ.

Press, Cambridge, UK (1995).

[2] E. Bareiss, Sylvester's Identity and multistep integer-preserving Gaussian elimination, *Math. Comp.* **22**(103) (1968) 565-578.

[3] B. Beckermann, Zur Interpolation mit polynomialen Linearkombinationen beliebiger Funktionen, Thesis, Univ. Hannover, 1990.

[4] B. Beckermann, The structure of the singular solution table of the M-Padé approximation problem, *J. Comput. Appl. Math.* **32** (1 & 2)(1990) 3-15.

[5] B. Beckermann, A reliable method for computing M-Padé approximants on arbitrary staircases, *J. Comput. Appl. Math.* **40** (1992) 19-42.

[6] B. Beckermann, The stable computation of formal orthogonal polynomials, *Numerical Algorithms* **11** (1996) 1-23.

[7] B. Beckermann & G. Labahn, A uniform approach for Hermite Padé and simultaneous Padé Approximants and their matrix generalizations, *Numerical Algorithms* **3** (1992) 45-54.

[8] B. Beckermann & G. Labahn, A uniform approach for the fast, reliable computation of Matrix-type Padé approximants, *SIAM J. Matrix Anal. Appl.* **15** (1994) 804-823.

[9] B. Beckermann & G. Labahn, Recursiveness in Matrix Rational Interpolation Problems, *J. Comput. Appl. Math.* **77** (1997) 5-34.

[10] B. Beckermann, S. Cabay & G. Labahn, Fraction–free Computation of Matrix Padé Systems, *Proceedings of ISSAC'97*, Maui, ACM Press, (1997) 125-132.

[11] R.R. Bitmead, S.Y. Kung, B.D.O. Anderson & T. Kailath, Greatest Common Divisors via Generalized Sylvester and Bezout Matrices, *IEEE Trans. Automat. Contr.* **AC-23** (1978) 1043-1046.

[12] A.W. Bojanczyk, R.P. Brent & F.R. de Hoog, Stability analysis of a general Toeplitz systems solver, *Numerical Algorithms* **10** (1995) 225-244.

[13] A.W. Bojanczyk, R.P. Brent, F.R. de Hoog & D.R. Sweet, On the Stability of the Bareiss and related Toeplitz Factorization Algorithms, *SIAM J. Matrix Anal. Appl.* **16** (1995) 40-57.

[14] R. Brent, F.G. Gustavson and D.Y.Y. Yun, Fast solution of Toeplitz systems of equations and computation of Padé approximants, *J. of Algorithms* **1** (1980) 259-295.

[15] W. Brown & J.F. Traub, On Euclid's algorithm and the theory of subresultants, *J. ACM* **18** (1971) 505-514.

[16] A. Bultheel & M. Van Barel, A matrix Euclidean Algorithm and the Matrix Minimal Padé Approximation Problem, in: *Continued Fractions and Padé Approximants*, C. Brezinski, ed., Elsevier, North–Holland (1990).

[17] S. Cabay and D.K. Choi, Algebraic computations of scaled Padé fractions, *SIAM J. of Computing*, **15** (1986), 243-270.

[18] S. Cabay, G. Labahn & B. Beckermann, On the Theory and Computation of Non-perfect Padé-Hermite Approximants, *J. Comput. Appl. Math.* **39** (1992) 295-313.

[19] S. Cabay, A. R. Jones & G. Labahn, Computation of Numerical Padé-Hermite and Simultaneous Padé Systems II: A Weakly Stable Algorithm, *SIAM J. Matrix Anal. Appl.* **17** (1996) 268-297.

[20] S. Cabay & P. Kossowski, Power Series remainder sequences and Padé fractions over an integral domain, *J. Symbolic Computation*, 10 (1990), pp. 139-163.

[21] S. Cabay & G. Labahn, A superfast algorithm for multidimensional Padé systems, *Numerical Algorithms* **2** (1992) 201-224

[22] S. Cabay & R. Meleshko, A weakly stable Algorithm for Padé Approximants and the Inversion of Hankel matrices, *SIAM J. Matrix Anal. Appl.* **14** (1993) 735-765.

[23] S. Chandrasekaran & A.H. Sayed, Stabilizing the generalized Schur algorithm, *SIAM J. Matrix Anal. Appl.* **14** (1996) 950-983.

[24] G. Collins, Subresultant and Reduced Polynomial Remainder Sequences. *J. ACM* **14**, (1967) 128-142

[25] S.R. Czapor and K.O. Geddes, A comparison of algorithms for the symbolic computation of Padé approximants. *Proceedings of EUROSAM'84*, J. Fitch (ed.), Lecture Notes in Computer Science, No. 174, Springer-Verlag, Berlin, 1984, pp. 248-259.

[26] R.W. Freund & H. Zha, Formally biorthogonal polynomials and a look–ahead Levinson algorithm for general Toeplitz systems, *Linear Algebra Appl.* **188/89** (1993) 255-303.

[27] R.W. Freund & H. Zha, A look–ahead algorithm for the solution of general Hankel systems, *Numer. Math.* **64** (1993) 295-321.

[28] W.F. Ford & A. Sidi, An Algorithm for a Generalization of the Richardson Extrapolation Process, *SIAM J. Numer. Anal.* **24** (1987) 1212-1232.

[29] K.O. Geddes, S.R. Czapor & G. Labahn, *Algorithms for Computer Algebra*, (Kluwer, Boston, MA, 1992)

[30] I. Gohberg, T. Kailath & V. Olshevski, Fast Gaussian elimination with partial pivoting for matrices with displacement structure, *Math. Comp.* **64** (1995) 1557-1567.

[31] G. Golub & V. Olshevski, Pivoting for structured matrices, with Applications. Manuscript (1997). `http://www-isl.stanford.edu/~olshevsk`

[32] M. Gu, Stable and efficient algorithms for structured systems of linear equations, to appear in *SIAM J. Matrix Anal. Appl.*

[33] M.H. Gutknecht, Stable Row Recurrences for the Padé Table and Generically Superfast Look-ahead Solvers for Non-Hermitian Toeplitz Systems, *Linear Algebra Appl.* **188/89** (1993) 351-421.

[34] M.H. Gutknecht & M. Hochbruck, Look-ahead Levinson and Schur algorithms for non–Hermitian Toeplitz Systems, *Numer. Math.* **70** (1995) 181-227.

[35] G. Heinig & K. Rost, *Algebraic methods for Toeplitz-like matrices and operators*, Operator Theory, v13, (Birkhäuser, Basel, 1984).

[36] T. Kailath, Linear systems, Prentice–Hall (1980).

[37] D. Knuth, The Art of Computer Programming Vol 2, Addison-Wesley, (1981)

[38] S.Y. Kung, T. Kaliath & M. Morf, A generalized resultant matrix for polynomial matrices, in: *Proc. IEEE Conf. on Decision and Control*, Florida (1976) 892-895.

[39] G. Labahn, Inversion Components of Block Hankel-like Matrices, *Linear Algebra Appl.* **177** (1992) 7-48

[40] G. Labahn, Inversion Algorithms for Rectangular-block Hankel Matrices, Research Report CS-90-52 (1990), Univ. of Waterloo.

[41] G. Labahn & S. Cabay, Matrix Padé fractions and their computation, *SIAM J. of Computing* **18** (1989) 639-657.

[42] G. Labahn, D.K. Choi and S. Cabay, Inverses of Block Hankel and Block Toeplitz Matrices, *SIAM J. of Computing* **19** (1990) 98-123.

[43] Z. Li, A Subresultant Theory for Linear Differential, Linear Difference and Ore Polynomials, with Applications, PhD Thesis, Univ. Linz, Austria, 1996.

[44] W. Lübbe, Über ein allgemeines Interpolationsproblem - Lineare Identitäten zwischen benachbarten Lösungssystemen, Thesis, Univ. Hannover, 1983.

[45] K. Mahler, Perfect systems, *Compos. Math.* **19** (1968) 95-166.

[46] S. Paszkowski, Quelques Algorithmes de l'Approximation de Padé–Hermite, Publication ANO 89, Univ. Lille 1 (1982).

[47] S. Paszkowski, Recurrence relations in Padé-Hermite approximation, *J. Comput. Appl. Math.* **19** (1987) 99-107.

[48] S. Paszkowski, Hermite Padé approximation: basic notions and theorems. *J. Comput. Appl. Math.* **32** (1990) 229-236.

[49] B. Salvy & P. Zimmermann, Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable, *ACM Transactions on Mathematical Software (TOMS)*, 20(2) (1994) 163-177.

[50] A. Sidi, On a Generalization of the Richardson Extrapolation Process, *Numer. Math.* **57** (1990) 365-377.

[51] M. Van Barel & A. Bultheel, The computation of non-perfect Padé-Hermite approximants, *Numerical Algorithms* **1** (1991) 285-304.

[52] M. Van Barel & A. Bultheel, A general module theoretic framework for vector M-Padé and matrix rational interpolation, *Numerical Algorithms* **3** (1992) 451-462.

[53] M. Van Barel & A. Bultheel, A look–ahead algorithm for the solution of block Toeplitz systems, Report TW 224, Katholieke Universiteit Leuven, (1995), to appear in *Linear Algebra Appl.*

[54] M. Van Hoeij, Factorization of Differential Operators with Rational Function Coefficients. *Journal of Symbolic Computation* (1998).

[55] G. Villard, Computing Popov and Hermite forms of polynomial matrices, *Proceedings of ISSAC'96* Zurich, ACM Press (1996) 250-258.