Fraction-free Computation of Matrix Padé Systems

Bernhard Beckermann,
Laboratoire d'Analyse Numérique et d'Optimisation,
UFR IEEA – M3, USTL Flandres Artois,
F–59655 Villeneuve d'Ascq CEDEX, France,
e-mail: bbecker@ano.univ-lille1.fr
Stan Cabay,
Department of Computer Science,
University of Alberta, Edmonton, Alberta, Canada,
e-mail: cabay@cs.ualberta.ca
and
George Labahn
Department of Computer Science,
University of Waterloo, Waterloo, Ontario, Canada,
e-mail: glabahn@daisy.uwaterloo.ca

Abstract

We present a fraction-free approach to the computation of matrix Padé systems. The method relies on determining a modified Schur complement for the coefficient matrices of the linear systems of equations that are associated to matrix Padé approximation problems. By using this modified Schur complement for these matrices we are able to obtain a fast hybrid fraction-free algorithm for their computation. The algorithm is general and requires no extra assumptions on its input.

1 Introduction

Let A be a $s \times s$ matrix power series. A matrix Padé approximant of type (p, q) for A is a pair of $s \times s$ matrix polynomials U and V having degrees bounded by m and n, respectively, which satisfy

$$A(z) \cdot U(z) - V(z) = O(z^{p+q+1}).$$

Technically speaking the above defines a right matrix Padé approximant with a left approximant defined in the obvious way. In the scalar case this roughly means that $A(z) \approx V(z) \cdot U(z)^{-1}$, at least for the first $p + q + 1$ terms. In the matrix case there are additional problems since even forming such a rational expression may be impossible due to the singularity of $U(z)$. Matrix Padé approximants appear in such applications as solving systems of linear equations having block Hankel or Toeplitz matrices [17], digital filtering theory [9] and in the realization of linear feedback control systems from input-output data [18]. In computer algebra they are of interest because of their close relationship with gcd and matrix gcd problems [11] and their use in the second step of the Block Wiedemann Algorithm. Additional applications are noted in [1].

Existing fast algorithms for the computation of matrix Padé approximants such as [4, 16] all assume that the coefficient matrices have entries from a field with the cost of a single arithmetic operation a constant. In particular, these algorithms do not take into consideration problems such as growth of coefficients in intermediate operations.

In this paper we consider the problem of solving matrix Padé approximation problems where the coefficient matrices of our power series have entries from an integral domain rather than a field. It is the purpose of this paper to compute this approximation problem by the use of algorithms which keep the growth of coefficients down to a reasonable size. We do this by determining a known common divisor at each recursive step of the computation. In this way our process is similar to fraction-free algorithms for solving linear systems [2], computation of scalar greatest common divisors [8, 13] and scalar Padé approximation [11].

Our methods study the linear systems that are associated to matrix Padé problems. These matrices have the structure of block Sylvester matrices. By looking at the associated linear systems we are able to obtain recursive algorithms that control the size of their output by predicting common divisors along the recursive path. The main tool that we use is a modified Schur complement for the coefficient matrices that appear in our rational approximation problems. In all cases our algorithms are fast. In most cases they are asymptotically at least an order of magnitude faster than any known alternative fraction free method.

Our results have applications beyond the computation of matrix rational approximations. For example, in the case of structured matrix inversion, we provide fraction-free solvers for linear systems having coefficient matrices with the structure of block Hankel or Toeplitz matrices [15].

We consider the method that is exploited here, that of using a modified Schur complement on a structured system of equations, to be a very general method for constructing fraction-free algorithms for *matrix-like* rational approximation and interpolation problems. These matrix-like approximation problems include Hermite-Padé, simultaneous Padé and multi-point Padé approximation problems along with their vector and matrix generalizations [1, 4]. Such computations appear in such diverse applications as the Gfun package of Salvy and Zimmerman [19] for determining recurrences relations, factorization of linear differential operators [23], computation of matrix normal forms [24], inversion of structured matrices [15] and computation of common divisors of matrix greatest common divisors [7].

The remainder of this paper is organized as follows. In Section 2 we introduce the main building blocks of our algorithms, Mahler Matrix Padé systems, along with their associated linear systems. In Section 3 we consider computing such systems along an off-diagonal path and show how to break the problem of computing one system into two smaller systems via a modified Schur complement. Section 4 gives a fast fraction-free hybrid algorithm for their computation while the following section provides a complexity analysis of the algorithm. Section 6 interprets the computation in terms of matrix power series residual sequences and shows the relationship between related computations, in particular the scalar subresultant gcd algorithm. The last section includes a conclusion along with a discussion of future research directions.

Finally, we mention that most of the proofs in this paper are simply short sketches. Complete proofs will be given in a subsequent article which tackles the fraction-free computation of more general matrix-like rational approximants.

2 Mahler Matrix Padé Systems

Let A and B be matrix power series

$$A(z) = \sum_{i=0}^{\infty} a_i z^i \text{ and } B(z) = \sum_{i=0}^{\infty} b_i z^i$$

with the a_i and b_i square matrices of size $s \times s$. We assume that the entries of our matrices come from an integral domain $\mathbf{D}$ and that $[a_0, b_0]$ is of full rank. This in fact is not a strong restriction since our main application, matrix Padé approximation, has $B(z) = -I$, the identity.

Definition 2.1 (Matrix Padé Approximants)
Let (p, q) be a pair of integers. A right matrix Padé approximant *for (A, B) of type (p, q) is a pair U, V of matrix polynomials, having degrees at most p and q, respectively, such that*

$$A(z) \cdot U(z) + B(z) \cdot V(z) = z^{p+q+1} W(z) \qquad (1)$$

with W a matrix power series. □

The matrix power series W is called the residual. When $B = -I$ one obtains the classic matrix Padé approximant for A. There is also a corresponding definition of a left matrix Padé approximant. A matrix Padé approximant of a given type always exists [16, Theorem 2.2]. Technically, we require that our domain be a field, however one can always work with quotient fields and then clear denominators to ensure existence for integral domains.

The primary tool that we use for the recursive computation of a Matrix Padé Approximant is a pair of such approximants organized in a matrix.

Definition 2.2 (Mahler Matrix Padé Systems (MMPS))
A (Right) Mahler Matrix Padé System *for (A, B) of type (p, q) is a block matrix polynomial*

$$\mathbf{P} = \begin{bmatrix} S & U \\ T & V \end{bmatrix}$$

with (S, T) an approximant of type $(p, q-1)$ and (U, V) an approximant of type $(p-1, q)$. □

For our purposes, we are interested in those MMPS which have certain added invertibility conditions. In particular, we are interested in the case where the leading coefficients of both S and V are invertible. When both leading coefficients are the identity we call this a monic normalization [6, Section 4.1]. However, since we are interested in fraction-free computation we will weaken this requirement and ask that the coefficients of both S and V be a (same) nonzero constant times the identity and call it a *nearly-monic normalization.* The nonzero constant will be referred to as a *normalization constant* of the system.

Remark 2.3 *A different type of Padé system is used in the hybrid algorithms of [4, 15, 16, 17]. A* Matrix Padé System *of type (p, q), has the first block column a matrix Padé approximant of type $(p-1, q-1)$ and the second block column is a matrix Padé approximant of type (p, q). The normalization used in that case was that the residual, R of the first block and the denominator, V, of the second block both have nonsingular trailing matrices (called a co-monic normalization [6, Section 4.1]).*

It is not true that a nearly-monic MMPS always exists. In order to determine when such a system in fact does exist, we study the associated linear system. This linear system is specified by the restrictions on the order of the products in (1), on the degrees and finally also the particular highest coefficients.

Let

$$C_{p,q} = \left[\begin{array}{ccc|ccc} a_0 & & & b_0 & & \\ \vdots & \ddots & & \vdots & \ddots & \\ & & a_0 & & & \\ & & & & & b_0 \\ \vdots & & & \vdots & & \\ a_{p+q-1} & \cdots & a_{q-1} & b_{p+q-1} & \cdots & b_{p-1} \\ \hline 0 & \cdots\ 0 & I & 0 & \cdots & 0 \\ 0 & & 0 & 0 & \cdots\ 0 & I \end{array}\right].$$

We will use the notation $C_{p,q}(A, B)$ when we need to explicitly mention the dependence of this matrix on the power series A and B.

Equating coefficients of (1) implies that a matrix Padé approximant of type (p, q) can be build by solving the linear system

$$T_{p,q} \cdot X = 0$$

for s linear independent solutions where $T_{p,q}$ denotes the block Sylvester matrix defined by the first $p+q$ block rows of $C_{p,q}$.

Since the system has s more variables than equations such a solution can always be constructed. Similarly, a nearly-monic MMPS with normalization constant $c \neq 0$ exists if and only if we can solve the block system

$$C_{p,q} \cdot P = c \cdot E \qquad (2)$$

where

$$P = \begin{bmatrix} s_0 & u_0 \\ \vdots & \vdots \\ s_p & u_p \\ t_0 & v_0 \\ \vdots & \vdots \\ t_q & v_q \end{bmatrix} \text{ and } E = \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ \vdots & \vdots \\ 0 & 0 \\ I & 0 \\ 0 & I \end{bmatrix},$$

where here and in the sequel I denotes the identity matrix of size s.

Clearly this is the case when $C_{p,q}$ is nonsingular. We also know from [15, Theorem 3.4] that the converse is true. This is summarized in the following

Theorem 2.4 *A nearly-monic MMPS of type (p, q) exists if and only if the matrix $C_{p,q}$ is nonsingular. A nearly-monic MMPS with a given normalization constant is unique.* □

Remark 2.5 *In the case of Matrix Padé Systems, the corresponding statement is that a co-monic system exists if and only if its associated block Sylvester matrix is nonsingular [4, 15, 16].*

Example 2.6 *Let A be the matrix power series having the first 9 terms as*

$$\begin{bmatrix} 1+z+3z^2+4z^4+7z^7 & 2z+z^5+z^6 \\ 2-2z-3z^3+z^6-z^8 & 2z^2+3z^3+z^4-3z^5+5z^6 \end{bmatrix}$$

and $B = -I$. Using fraction-free Gaussian elimination applied to the linear system (2) with $p = 1$ and $q = 2$ gives a nearly monic MMPS

$$\mathbf{P}(z) = \begin{bmatrix} -2+6z & -4 & 2 & 0 \\ 6 & 6z & 0 & 3 \\ -2+16z & -4-4z & 2+2z+6z^2 & 6z \\ -4+16z & -8+8z & 4-4z & 6z^2 \end{bmatrix}$$

having normalization constant 6. Notice that there is no nearly monic MMPS with a smaller normalization constant.

The first 3 terms of the residuals are then given by

$$R(z) = \begin{bmatrix} 18-8z+30z^2 & -16z \\ 24-12z-18z^2 & 24+18z+6z^2 \end{bmatrix} + O(z^3)$$

and

$$W(z) = \begin{bmatrix} 8z & 3z^2 \\ -6 & 9+3z-9z^2 \end{bmatrix} + O(z^3).$$

□

Example 2.7 *In some cases one can derive formulas for a MMPS of a given type directly by solving the linear system of equations. For example, let A and B be scalar power series with nonzero leading terms a_0 and b_0. For any positive integer k we see that $det(C_{k,0}) = a_0^k$ so that a MMPS of type $(k, 0)$ exists. Solving the linear system (2) gives*

$$\begin{bmatrix} a_0^k \cdot z^k & -q(z) \\ 0 & a_0^k \end{bmatrix}$$

as a nearly-monic MMPS of type $(k, 0)$ where $q(z) = q_0 + \cdots + q_{k-1}z^{k-1}$ solves the linear equation

$$\begin{bmatrix} a_0 & & \\ \vdots & \ddots & \\ a_{k-1} & \cdots & a_0 \end{bmatrix} \cdot \begin{bmatrix} q_0 \\ \vdots \\ q_{k-1} \end{bmatrix} = a_0^k \cdot \begin{bmatrix} b_0 \\ \vdots \\ b_{k-1} \end{bmatrix}.$$

In this case the normalization constant for the system is a_0^k, the determinant of $C_{k,0}$. We note that the polynomial q is closely related to the classical operation of pseudo-division [14]. Indeed q is the pseudo-quotient of $a_0z^m + a_1z^{m-1} + \cdots$ divided into $b_0z^{m+k-1} + b_1z^{m+k-2} + \cdots$ where $m \geq k$.

3 Off-diagonal Computation of Mahler Matrix Padé Systems

Let (p, q) be a pair of integers such that $C_{p,q}$ is nonsingular (we call such a point a *normal point* [16]). By Cramer's rule one can always find a (unique) fraction-free solution to (2) when $c = det(C_{p,q})$, which we call a *Cramer solution*. Besides being a fraction-free solution (i.e. elements from $\mathbf{D}$) of our rational approximation problem, a Cramer solution is known to have components of "reasonable size". Furthermore a Cramer solution is exactly the one determined when one uses fraction-free Gaussian elimination (c.f. [14]), and hence is computable with a well known bound on the growth of intermediate expressions. Example 2.7 gives a Cramer solution along with the associated nearly-monic MMPS of type $(k, 0)$ in the case of a pair of scalar power series.

Fraction-free Gaussian elimination, however, does not take advantage of the added structure of the matrix $C_{p,q}$. In this section we will describe a recurrence from one normal point to a later normal point, computing Cramer solutions at every step. One can combine this with fraction-free Gaussian elimination to "jump" over any singular locations and hence obtain a hybrid algorithm. The main problem that needs to be overcome is that the recurrence actually jump from one Cramer solution to another Cramer solution.

Suppose that (p, q) is a normal point and that $\mathbf{P}$ is a nearly monic MMPS that corresponds to a Cramer solution. Let R and W be the residual matrix power series satisfying

$$[A(z), B(z)] \cdot \mathbf{P}(z) = z^{p+q}[R(z), W(z)].$$

Let $\hat{U}, \hat{V}$ be a matrix Padé approximant of type (r, r) for the residuals (R, W). Then a matrix Padé approximant of type $(p + r, q + r)$ is given by

$$\begin{bmatrix} U \\ V \end{bmatrix} = \mathbf{P} \cdot \begin{bmatrix} \hat{U} \\ \hat{V} \end{bmatrix}.$$

Thus one can compute a matrix Padé approximant of a certain type by computing the closest MMPS along an off-diagonal path and then a matrix Padé approximant of the residuals.

Suppose further that (r, r) is a normal point for the pair (R, W) with $\hat{\mathbf{P}}$ the corresponding MMPS for the residuals.

Then the product $\mathbf{P} \cdot \hat{\mathbf{P}}$ is a nearly monic MMPS and so $(p+r, q+r)$ is a normal point for (A, B). In addition, if $\mathbf{P}^*$ is the system that corresponds to a Cramer solution of type $(p+r, q+r)$ then by Theorem 2.4 we have that

$$d \cdot \mathbf{P}^* = \mathbf{P} \cdot \hat{\mathbf{P}}$$

with d from $\mathbf{D}$ or its quotient field. We obtain a fraction-free recursion by determining the constant d in advance.

Let the components of $\mathbf{P}$ be given by

$$\mathbf{P} = \left[\begin{array}{cc} S & U \\ T & V \end{array} \right]$$

and set M to be the $(p+q+2r+2) \cdot s$ square matrix with block structure given by

$$M = \left[\begin{array}{ccc|ccc|ccc|ccc}
I & & & & & & s_0 & & & u_0 & & \\
& \ddots & & & & & \vdots & \ddots & & \vdots & \ddots & \\
& & I & & & & s_p & & s_0 & u_p & & u_0 \\
& & & & & & & \ddots & \vdots & & \ddots & \vdots \\
& & & & & & & & s_p & & & u_p \\
\hline
& & & I & & & t_0 & & & v_0 & & \\
& & & & \ddots & & \vdots & \ddots & & \vdots & \ddots & \\
& & & & & I & t_q & & t_0 & v_q & & v_0 \\
& & & & & & & \ddots & \vdots & & \ddots & \vdots \\
& & & & & & & & t_q & & & v_q \\
\multicolumn{3}{c}{\underbrace{\qquad\qquad}_{p}} & \multicolumn{3}{c}{\underbrace{\qquad\qquad}_{q}} & \multicolumn{3}{c}{\underbrace{\qquad\qquad}_{r+1}} & \multicolumn{3}{c}{\underbrace{\qquad\qquad}_{r+1}}
\end{array} \right]$$

Theorem 3.1 (Modified Schur Complement) *Let $\hat{C}_{r,r} = C_{r,r}(R, W)$, with (R, W) being the residual matrix power series of the MMPS of type (p, q). Then*

$$C_{p+r,q+r} \cdot M = \left[\begin{array}{ccccc} I & & & & \\ & \ddots & & & \\ & & I & & \\ & & & c \cdot I & \\ & & & & c \cdot I \end{array} \right] \cdot \left[\begin{array}{cc} \bar{C}_{p,q} & 0 \\ \# & \hat{C}_{r,r} \end{array} \right] \tag{3}$$

where $C_{p+r,q+r} = C_{p+r,q+r}(A, B)$, $C_{p,q} = C_{p,q}(A, B)$ and $\bar{C}_{p,q}$ a block matrix with determinant determined by $c = det(C_{p,q}) = (-1)^{qs^2} \cdot det(\bar{C}_{p,q})$.

Furthermore, M is nonsingular with

$$det(M) = (-1)^{(r+1)s^2} \cdot c^{2s(r+1)},$$

and

$$det(C_{p+r,q+r}) \cdot [det(C_{p,q})]^{2sr-1} = (-1)^{(q+r+1)s^2} det(\hat{C}_{r,r}). \tag{4}$$

Proof: The first part of the theorem follows from multiplying out the two matrices on the left of equation (3) and noting the order conditions defining the MMPS's. The upper left hand corner block matrix with block sizes $(p+q) \times (p+q)$ is the same as $C_{p,q}$ after removing the last two block rows and the block columns $p+1$ and $p+1+q+1$. According to the particular structure of the last two block columns we have $det(\bar{C}_{p,q}) = (-1)^{qs^2} \cdot det(C_{p,q})$.

The second part of the theorem follows from reducing the determinant of M (by expanding along identity row and columns) to that of the determinant of

$$\left[\begin{array}{cccccc}
s_p & \cdots & & u_p & \cdots & \\
& \ddots & \vdots & & \ddots & \vdots \\
& & s_p & & & u_p \\
t_q & \cdots & & v_q & \cdots & \\
& \ddots & \vdots & & \ddots & \vdots \\
& & t_q & & & v_q
\end{array} \right]$$

times $(-1)^{(r+1)s^2}$.

Since $t_q = u_p = 0$ and $s_p = v_q = c \cdot I$ the determinant of the latter matrix is given by $c^{2s(r+1)}$. A proof of (4) follows immediately by taking determinants in (3). □

Suppose that $det(C_{p,q}) \neq 0$, and denote by P the corresponding Cramer solution of type (p, q). Let (R, W) be the residual matrix power series for $\mathbf{P}$, the corresponding MMPS. We wish to compute the "next" existing Cramer solution P^* of type $(p+r, q+r)$, $r \geq 1$. From Theorem 3.1 we know that $det(\hat{C}_{r,r})$ and $det(C_{p+r,q+r})$ only vanish simultaneously. Thus, according to Theorem 2.4, it is sufficient to find the smallest $r > 0$ with $\hat{C}_{r,r}$ being nonsingular.

Let $\hat{P}$ be the Cramer solution for the linear system $\hat{C}_{r,r}$ determined by the residual matrix power series (R, W). Let

$$\bar{P} = M \cdot \left[\begin{array}{c} 0 \\ \hat{P} \end{array} \right] \tag{5}$$

and $\hat{c} = det(\hat{C}_{r,r})$. By Theorem 3.1 we see that $\bar{P}$ is a solution to equation (2) with normalization constant

$$c^{(2s+1)} \cdot \hat{c} = det(C_{p+r,q+r}) \cdot det(M).$$

Hence $c^{2s} \cdot \bar{P}$ is $det(M)$ times the Cramer solution of type $(p+r, q+r)$. In terms of the associated nearly-monic MMPS we have, by uniqueness,

$$det(M) \cdot \mathbf{P}^* = \pm c^{2s} \cdot \mathbf{P} \cdot \hat{\mathbf{P}}. \tag{6}$$

Theorem 3.2 *Let $det(\hat{C}_{r,r}) \neq 0$, and denote by $\hat{\mathbf{P}}$ the MMPS for the corresponding Cramer solution of type (r, r). Then*

$$(-1)^{(q+r+1)s^2} [det(C_{p,q})]^{2sr} \cdot \mathbf{P}^* = \mathbf{P} \cdot \hat{\mathbf{P}}.$$

Proof: Define $\bar{\mathbf{P}} := \mathbf{P} \cdot \hat{\mathbf{P}}$. It is easily checked that $\bar{\mathbf{P}}$ is a nearly–monic MMPS of order $(p+r, q+r)$, and thus $\bar{\mathbf{P}} = d \cdot \mathbf{P}^*$ by uniqueness. The constant d may be obtained by comparing the leading coefficients with the help of formula (4) from Theorem 3.1. □

In terms of MMPS one can obtain the system $\mathbf{P}^*$ by first computing the MMPS for the smaller problems, multiplying the two matrix polynomials and then dividing out by the known common factor $\pm c^{2sr}$.

4 A Fraction-free Matrix Padé Algorithm

The discussion following Theorem 3.1 along with Theorem 3.2 provides a recursive algorithm for computing a nearly monic MMPS that corresponds to determining a Cramer solution in terms of previous Cramer solutions along an off-diagonal path:

FF Matrix Padé Algorithm
Given: $p \leq q$, compute the MMPS at the last normal point along the path $(r, q-p+r)_{r=0,\ldots,}$.

- [Initialization]
  For $r_0 = 1, \ldots$ up to a maximum of p, use fraction-free Gaussian elimination to find the first nonsingular matrix $C_{r_0, q-p+r_0}$ and the MMPS $\mathbf{P}_0$ of type $(r_0, q-p+r_0)$ for (A, B) associated to the Cramer solution of the corresponding linear system.
- [Iteration]
  While $p_i < p$ do: given a normal point (p_i, q_i) and MMPS $\mathbf{P}_i$ of type (p_i, q_i) (with $q_i - p_i = q - p$) for (A, B) associated to the Cramer solution of the corresponding linear system.
  - [Compute Residual Mahler System]
    For $r_i = 1, \ldots$ up to a maximum of $p - p_i$ compute r_i terms of the residual matrix power series. Use fraction-free Gaussian elimination to determine the first normal point (r_i, r_i) of the residual power series. Let $\hat{\mathbf{P}}_i$ be the MMPS associated to the Cramer solution of the corresponding linear system.
  - [Compute New Mahler System]
    Compute $\bar{\mathbf{P}} = \mathbf{P}_i \cdot \hat{\mathbf{P}}_i$ and c_i, the leading coefficient of $\mathbf{P}_i$. Divide each element of $\bar{\mathbf{P}}$ by the known common divisor c^{2sr_i} to obtain the new MMPS $\mathbf{P}_{i+1}$. This MMPS is the one associated to the Cramer solution of the corresponding linear system $(p_{i+1}, q_{i+1}) = (p_i + r_i, q_i + r_i)$. Set $i = i + 1$ and return to the iteration step.

□

We remark that for our purposes a leading coefficient of a matrix polynomial is the coefficient matrix that corresponds in row i to the n_i coefficient of z for n_i the maximum degree of entries in the i-th row.

Example 4.1 *We illustrate the FF Matrix Padé algorithm by giving an example of a single step of the iteration. Let* $\mathbf{D}$ *be the integers and* A, B *the matrix power series of Example 2.6. From this example, we have already obtained* $\mathbf{P}$*, a MMPS of type* $(1,2)$ *for* (A, B) *corresponding to the Cramer solution. In this case the normalization constant is* 6.

Working with the residual matrix power series and using fraction-free Gaussian elimination shows that $(1,1)$ *is the first normal point along the path* $(t,t)_{t=1,\ldots}$. *Then* $\hat{\mathbf{P}}$*, the MMPS corresponding to the Cramer solution of type* $(1,1)$ *for the residual is given by*

$$\begin{bmatrix} -18144\,z & 0 & 0 & 0 \\ 25272 & 31104 - 18144\,z & -7776 & 11664 \\ 91368 & 62208 & -15552 - 18144\,z & 23328 \\ -6480 & -41472 & 10368 & -15552 - 18144\,z \end{bmatrix}$$

Multiplying $\mathbf{P} \cdot \hat{\mathbf{P}}$ *and dividing out by the common factor* $6^4 = 1296$ *gives*

$$\begin{bmatrix} 63 + 28\,z - 84\,z^2 & 56\,z & -28\,z & 0 \\ -15 + 33\,z & -96 + 144\,z - 84\,z^2 & 24 - 36\,z & -36 + 12\,z \\ 63 + 61\,z + 199\,z^2 & -136\,z + 344\,z^2 & 20\,z - 100\,z^2 - 84\,z^3 & -72\,z + 24\,z^2 \\ 126 - 70\,z - 254\,z^2 & 112\,z - 304\,z^2 & -56\,z + 104\,z^2 & -72\,z^2 - 84\,z^3 \end{bmatrix}$$

the MMPS of type $(2,3)$ *for* (A, B) *corresponding to the Cramer solution. Note that the normalization constant in this case is* -84 *and hence the common divisor for the next step would be* 49787136 *(should* $(3,4)$ *be a normal point).*

□

For a given pair (A, B) our algorithm computes a sequence $\{\mathbf{P}_i\}_{i=1,\ldots}$ of MMPS of types $\{(p_i, q_i)\}_{i=1,\ldots}$. For each i there exists a pair of residual matrix power series (R_i, W_i) and a MMPS $\hat{\mathbf{P}}$ of type (r_i, r_i) with $r_i = q_{i+1} - q_i$ such that

$$[A(z), B(z)] \cdot \mathbf{P}_i(z) = z^{p_i + q_i} [R_i(z), W_i(z)] \tag{7}$$

$$[R_i(z), W_i(z)] \cdot \hat{\mathbf{P}}_i(z) = z^{2r_i} [\hat{R}_{i+1}(z), \hat{W}_{i+1}(z)] \tag{8}$$

$$c_i{}^{2r_i s} \cdot \mathbf{P}_{i+1}(z) = \mathbf{P}_i(z) \cdot \hat{\mathbf{P}}_i(z) \tag{9}$$

where $lcoeff(\mathbf{P}_i(z)) = c_i \cdot I_{2s}$.

Note that, because of the uniqueness of nearly-normal MMPS at a normal point we always have that

$$[\hat{R}_{i+1}(z), \hat{W}_{i+1}(z)] = c_i{}^{2r_i s} \cdot [R_{i+1}(z), W_{i+1}(z)]$$

for each $i \geq 1$. This allows us to interpret our computational process as a *matrix power series residual sequence.* By this we mean that our algorithm computes a sequence $\{\hat{\mathbf{P}}_i\}_{i=0,\ldots}$ of MMPS for a sequence of residuals (R_i, W_i) such that

$$[R_0(z), W_0(z)] = [A(z), B(z)],$$

with

$$[R_0(z), W_0(z)] \cdot \hat{\mathbf{P}}_0(z) = z^{p_0 + q_0} \cdot [R_1(z), W_1(z)]$$

and, for $i > 0$,

$$[R_i(z), W_i(z)] \cdot \hat{\mathbf{P}}_i(z) = z^{2r_i} \cdot \beta_i \cdot [R_{i+1}(z), W_{i+1}(z)]$$

with $\beta_i \in \mathbf{D}$. From equation (9) we have that, for each i, $\beta_i = c_i{}^{2r_i s}$ where $lcoeff(\mathbf{P}_i(z)) = c_i \cdot I_{2s}$. In terms of the sequence $\hat{\mathbf{P}}_i(z)$ the β_i are given by

$$\beta_0 := 1; \beta_{i+1} = \hat{c}_i \cdot \beta_i{}^{1-2r_i s} \text{ for } i \geq 1$$

where $lcoeff(\hat{\mathbf{P}}_i(z)) = \hat{c}_i \cdot I_{2s}$.

This interpretation allows one to view our computation in a similar light to that used for classical PRS type methods for computing polynomial gcd's.

Table 1: Size Estimates

Variables	Size Estimates
A, B	$O(\kappa)$
R, W	$O(s \cdot (p+q) \cdot \kappa)$
$\mathbf{P}$	$O(s(p+q) \cdot \kappa)$
$\hat{\mathbf{P}}$	$O(rs^2(p+q) \cdot \kappa)$
$\bar{\mathbf{P}}$	$O(rs^2(p+q) \cdot \kappa)$

5 Complexity

In this section, we derive asymptotic estimates for the cost of our fraction-free algorithm. We assume that classical methods are used for performing all arithmetic in $\mathbf{D}$. In addition, to keep the analysis tractable we assume that

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

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. Using this model, the cost of fraction-free Gaussian elimination of an $n \times n$ system is $\mathrm{O}(n^5 \cdot N^2)$ where N is the largest size of the coefficients of the system.

In terms of size, suppose that every term in both A and B is bounded by the constant κ. We obtain the size for $\mathbf{P}$ from the fact that its components are Cramer solutions and hence have sizes easily derived from Cramer's rule. The residuals R and W are computed from the original power series multiplied by $\mathbf{P}$ and hence their sizes are determined by the multiplications. Similarly, the system $\hat{\mathbf{P}}$ has size determined from the sizes of the residual series. Finally, the updated system has size again determined by the original power series. The sizes are included in Table 1.

In terms of complexity of arithmetic operations, the cost of each iteration of the FF Matrix Padé algorithm is the cost of generating the residual matrix power series along with the cost of fraction-free Gaussian elimination to determine the first normal point and compute the corresponding Cramer solution. Clearly the overall complexity depends on the largest "jump" required for one of these iterative steps.

Given an existing $\mathbf{P}$, we compute the first $2r$ terms of the matrix residuals by

$$\begin{array}{lcl} \text{cost}(R, W) & = & O(r \cdot s^2 \cdot \text{size}(\mathbf{P}) \cdot (s \cdot (p+q) \cdot \kappa)) \\ & = & O(r \cdot s^4 \cdot (p+q)^2 \cdot \kappa^2). \end{array}$$

The cost of fraction-free Gaussian elimination [2], used to solve (2), is then given by

$$\begin{array}{lcl} \text{cost}(\hat{\mathbf{P}}) & = & O(r^5 \cdot s^5 \cdot \text{size}^2(R, W)) \\ & = & O(r^5 \cdot s^7 \cdot (p+q)^2 \cdot \kappa^2). \end{array}$$

The remaining costs of the other computations are summarized in Table 2.

Table 2: Cost Estimates

Variables	Cost Estimates
R, W	$O(rs^4 \cdot (p+q)^2 \cdot \kappa^2)$
$\hat{\mathbf{P}}$	$O(r^5 s^7 (p+q)^2 \cdot \kappa^2)$
$\bar{\mathbf{P}}$	$O(r^2 s^6 (p+q)^3 \cdot \kappa^2)$
$\mathbf{P}$	$O(r^2 s^6 (p+q)^3 \cdot \kappa^2)$

To obtain a cost estimate of the algorithm, we sum the estimates given in Table 2 over all iterations. The algorithm computes a sequence of jumps $r_0, \ldots, r_k$ such that the final MMPS occurs with degree bounds (p_F, q_F) with

$$p_F = \sum_{i=0}^{k} r_i, q_F = (q-p) + \sum_{i=0}^{k} r_i.$$

If we assume that our step sizes are bounded by a small value (this is the case in most applications - indeed in most cases the assumption is made that $r_i = 1$ for all i (called a *normal* case)). In such cases the algorithm has a total cost estimate of

$$O(\sum_{i=0}^{k} r_i^2 s^6 (p_i + q_i)^3 \kappa^2) = O(s^6 (p_F + q_F)^4 \cdot \kappa^2).$$

For large p_F, q_F this compares with fraction-free Gaussian elimination which has a cost estimate of $O(s^5(p_F+q_F)^5 \cdot \kappa^2)$ for solving the linear system.

6 Scalar Mahler Systems

In the case of scalar power series our algorithm requires $O((p_F+q_F)^4 \cdot \kappa^2)$ arithmetic operations, an order of magnitude improvement over fraction-free Gaussian elimination. However, we have already seen in some cases that it is possible to replace elimination with pseudo-division, reducing the cost of a substep by an order of magnitude. Example 2.7 illustrates this for systems of type $(k, 0)$ when A begins with a non-zero coefficient. A similar formula holds for systems of type $(0, k)$ when B begins with a non-zero coefficient. In this section we show that in the scalar case we can replace elimination with pseudo-division in all the steps of our algorithm.

Lemma 6.1 *Let A and B be scalar power series with $a_0 \neq 0$ and $b_0 = \cdots = b_{k-1} = 0, b_k \neq 0$ with $k \geq 1$. Then (k, k) is a normal point for (A, B) and $\mathbf{P}$, the MMPS of type (k, k) associated to the corresponding Cramer solution, is*

$$\mathbf{P}(z) = (-a_0)^k \cdot \begin{bmatrix} b_k^k \cdot z^k & 0 \\ -q(z) & b_k^k \cdot z^k \end{bmatrix} \tag{10}$$

where $q(z) = q_0 + q_1 z + \cdots + q_{k-1} z^{k-1}$ satisfies

$$\begin{bmatrix} b_k & & \\ \vdots & \ddots & \\ b_{2k-1} & \cdots & b_k \end{bmatrix} \cdot \begin{bmatrix} q_0 \\ \vdots \\ q_{k-1} \end{bmatrix} = b_k^k \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_{k-1} \end{bmatrix}. \tag{11}$$

Proof: The determinant of $C_{k,k}$ is easily seen to be $(-a_0 \cdot b_k)^k$ so (k, k) is a normal point. The MMPS in equation (10) is determined directly by solving the associated linear system of equations. □

We note that the polynomial q is the pseudo-quotient of $b_k z^m + b_{k+1} z^{m-1} + \cdots$ divided into $a_0 z^{m+k-1} + a_1 z^{m+k-2} + \cdots$ where $m \geq k$.

A similar formula exists for a MMPS of type (k, k) when A and B are scalar power series with $a_0 = \cdots = a_{k-1} = 0, a_k \neq 0$ $(k \geq 1)$ and $b_0 \neq 0$. It is also straightforward to see that these give the first normal point along the diagonal.

A formula for determining the first normal point of type (k, k) when both A and B start with nonzero terms requires a bit more effort. Note that if one uses fraction-free Gaussian elimination on the *columns* of $C_{k,k}$ (which, given the structure of these matrices, is more natural than trying row elimination) then the first step is to form the coefficients in the series $b_0 A - a_0 B$. It turns out that the number of terms eliminated in this single step determines the location of the first normal point. Indeed we have

Lemma 6.2 *Let A and B be two power series and $a_0 \neq 0$ and $b_0 \neq 0$. Let H be the power series determined by*

$$b_0 \cdot A(z) - a_0 \cdot B(z) = z^k H(z).$$

Then (k, k) is a normal point of (A, B) and $\mathbf{P}$, the MMPS of type (k, k) associated to the corresponding Cramer solution is

$$\mathbf{P}(z) = \begin{bmatrix} h_0^k \cdot z^k - b_0 \cdot q(z) & -b_0 \cdot \bar{q}(z) \\ a_0 \cdot q(z) & h_0^k \cdot z^k + a_0 \cdot \bar{q}(z) \end{bmatrix} \tag{12}$$

where $q(z) = q_0 + q_1 z + \cdots + q_{k-1} z^{k-1}$ satisfies

$$\begin{bmatrix} h_0 & & \\ \vdots & \ddots & \\ h_{k-1} & \cdots & h_0 \end{bmatrix} \cdot \begin{bmatrix} q_0 \\ \vdots \\ q_{k-1} \end{bmatrix} = h_0^k \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_{k-1} \end{bmatrix} \tag{13}$$

and $\bar{q}(z) = \bar{q}_0 + \bar{q}_1 z + \cdots + \bar{q}_{k-1} z^{k-1}$ satisfies

$$\begin{bmatrix} h_0 & & \\ \vdots & \ddots & \\ h_{k-1} & \cdots & h_0 \end{bmatrix} \cdot \begin{bmatrix} \bar{q}_0 \\ \vdots \\ \bar{q}_{k-1} \end{bmatrix} = h_0^k \cdot \begin{bmatrix} b_0 \\ \vdots \\ b_{k-1} \end{bmatrix}. \tag{14}$$

Proof: The determinant of $C_{k,k}$ is easily seen to be h_0^k so (k, k) is a normal point. The MMPS in equation (12) is determined directly by solving the associated linear system of equations. That one has a Mahler system can also be verified by simply multiplying $[A(z), B(z)] \cdot \mathbf{P}(z)$ and noting that one has the correct order conditions. □

We note that both polynomials q and $\bar{q}$ are pseudo-quotients. In addition one easily verifies the link $b_0 \cdot q = a_0 \cdot \bar{q}$ so that in fact only one of the systems (13), (14), has to be solved explicitly.

Lemmas 6.1 and 6.2 along with Example 2.7 provide a faster way of computing all intermediate steps in the fraction-free algorithm of section 4. In this case we are replacing fraction-free Gaussian elimination by at most two pseudo-divisions at every step, reducing each of these intermediate steps by an order of magnitude. The resulting algorithm has a cost of $O((p_F + q_F)^3 \cdot \kappa^2)$. Finally, these results do not carry over to the matrix case since a non-zero element is not always invertible in the non-scalar case.

Example 6.3 *Let A and B be scalar power series given by*

$$A(z) = 1 + z^2 - 3\,z^4 - 3\,z^5 + 8\,z^6 + 2\,z^7 - 5\,z^8 + O(z^{14})$$

and

$$B(z) = 3 + 5\,z^2 - 4\,z^4 - 9\,z^5 + 21\,z^6 + O(z^{14}).$$

Then from Example 2.7 we determine that the MMPS of type $(0, 2)$ for (A, B) is given by

$$\mathbf{P}_1(z) = \begin{bmatrix} 9 & 0 \\ -3 & 9\,z^2 \end{bmatrix}$$

with residuals

$$R_1(z) = -6 - 15\,z^2 + 9\,z^4 + \cdots$$

and

$$W_1(z) = 27 + 45\,z^2 - 36\,z^4 + \cdots.$$

Computing $27R_1 + 6W_1$ gives

$$-135z^2 + 27z^4 + \cdots \tag{15}$$

hence the first normal point is at $(2, 2)$ for the residuals and so at $(2, 4)$ for the original power series. The MMPS of type $(2, 2)$ for the residuals is determined by pseudo-division of the power series (15) into R_1 and then W_1 and is given by

$$\hat{\mathbf{P}}_1(z) = \begin{bmatrix} -21870 + 18225\,z^2 & 98415 \\ -4860 & 21870 + 18225\,z^2 \end{bmatrix}.$$

Combining these and removing the common factor $9^4 = 6561$ gives the MMPS of type $(2, 4)$ for (A, B) as

$$\mathbf{P}_2(z) = \begin{bmatrix} -30 + 25\,z^2 & 135 \\ 10 - 15\,z^2 & -45 + 30\,z^2 + 25\,z^4 \end{bmatrix}$$

with residuals

$$R_2(z) = -45 + 35\,z^2 + 50\,z^3 - 125\,z^4 + \cdots$$

and

$$W_2(z) = 140 - 145\,z^2 - 225\,z^3 + 525\,z^4 + \cdots.$$

Forming $140R_2 + 45W_2$ gives

$$-1625z^2 - 3125z^3 + 6125z^4 + \cdots$$

As before the first normal point of the residuals is then at $(2, 2)$ and hence the next normal point is $(4, 6)$ for the original power series. Computing the MMPS for the residuals via Lemma 6.2, multiplying with the system at $(2, 4)$ and dividing by $25^4 = 390625$ gives the MMPS of type $(4, 6)$. Continuing on would determine that the next normal points are at $(5, 7)$ and $(6, 8)$. For these three normal points the normalization-constants for the corresponding nearly-monic systems are 169, -9326 and 260708. □

7 Conclusions

In this paper we have given an algorithm for the computation of nearly-monic MMPS for matrix power series having matrix coefficients from an integral domain. The algorithm is fraction-free in the sense that it does not require any arithmetic in quotient fields. In addition, the algorithm requires no extra conditions on the input and in most cases is at least an order of magnitude faster than existing fraction-free methods.

Fraction-free algorithms are useful for practical computation. They are also important as a necessary first step in developing other algorithms that control coefficient growth, in particular for developing algorithms based on modular reduction. We plan on using our theoretical results to build modular algorithms for matrix Padé approximants, with an expected additional order of magnitude improvement in complexity.

It is well-known that computing scalar Padé approximants along an off-diagonal path is closely related to computing greatest common divisors of two polynomials using the extended Euclidean algorithm [10, 11] (with the order of the coefficients of the polynomial reversed). Since our scalar algorithm computes Padé approximants along an off-diagonal path there is also a relation between our algorithm and fraction-free computation of greatest common divisors. We expect that our algorithm gives the subresultant gcd algorithm [8, 13] as a special case. One can compare Example 6.3 to a well-known gcd example first given by Knuth [14, Example7.6] as an example of a possible relationship between these two algorithms. A similar statement can be made regarding our algorithm and algorithms for fraction-free solving of Hankel systems [20] since the relationship between solving Hankel systems, computing Padé systems and the extended Euclidean algorithm is also well-known [21].

Finally, our algorithm computes matrix Padé approximants along an off-diagonal path in a hybrid way - recursing from normal point to normal point with the intermediate computations done via fraction-free Gaussian elimination. We expect that the algorithm described in this paper can be easily extended to compute along arbitrary paths, something that is not possible yet for the σ-bases algorithms of [4, 5]. This becomes significant in many applications where particular paths are known to consist of only normal points. For example, in the case of positive-definite block Toeplitz linear systems a vertical path has only normal points while nothing can be said about the corresponding off-diagonal path.

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

[5] 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.

[6] B. Beckermann & G. Labahn, Recursiveness in Matrix Interpolation Problems, To appear in *J. Comput. Appl. Math.* (1997) 29 pages.

[7] B. Beckermann & G. Labahn, Fraction-free Computation of Matrix GCD's and Rational Interpolants. Manuscript.

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

[9] A. Bultheel, Algorithms to compute the reflection coefficients of digital filters, *Numerical Methods of Approximation Thoery*, 7(1983)

[10] S. Cabay & D.K. Choi, Algebraic Computations of scaled Padé Fractions, *SIAM J. of Computing*, **6** (1986) 243-270.

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

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

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

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

[15] G. Labahn, Inversion Components of Block Hankel-like Matrices, *Linear Algebra and its Applications* **177** (1992) 7-48

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

[17] G. Labahn, D.K. Choi & S. Cabay, Inverse of Block Hankel and Block Toeplitz matrices, *SIAM J. of Computing*, 19(1) (1990) 98-123.

[18] H.H. Rosenbrock, *State-Space and Multivariable Theory.* New York: Wiley, 1970.

[19] 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.

[20] J.R. Sendra & J. Llovet, An Extended Polynomial GCD Algorithm using Hankel Matrices, *J. Symbolic Computation* **13**(1) (1992) 25-40.

[21] Y. Sugiyama, An Algorithm for Solving Discrete-time Wiener-Hopf Equations based on Euclid's Algorithm, *IEEE Trans. Inform. Theory* **32** (1986) 394-409.

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

[23] M. Van Hoeij, Factorization of Differential Operators with Rational Function Coefficients. To appear *Journal of Symbolic Computation* (1997).

[24] G. Villard, Personal Communication (1996)