A new approach to the rational interpolation problem : the vector case

Marc Van Barel and Adhemar Bultheel
Department of computer science
K.U. Leuven

January 2, 2000

Abstract

We generalize our earlier results on rational interpolation which were given in [19] for the scalar case and in [6] for the vector case when all the interpolation points coincide, to the case of vector data given at arbitrary points that may coincide or not. This is the vector valued Newton-Padé problem. We give a recursive algorithm which has the important advantage over other algorithms that we don't need a reordering of the given interpolation data to overcome a singularity in the interpolation table, not even in the non-normal vector case. It also generates all the information needed to give all solutions of the problem.

Keywords: Rational interpolation, Newton-Padé, vector interpolants, simultaneous rational approximation

1 Introduction

In an earlier paper [19], we showed that we could take away several drawbacks of the classical Newton-Padé rational interpolation problem by using a new approach to the rational interpolation problem (RIP). We also gave a recursive algorithm which has the important advantage over existing algorithms that we don't need a reordering of the given interpolation data. Our algorithm basically solves the linearized RIP and gives a solution that is in some clearly defined sense as simple as possible, but as a side product, it also generates all the information needed to give all solutions, not only for the linearized RIP but also for the proper RIP. When all the interpolation points coincide, we can reformulate the scalar RIP as a scalar minimal partial realization problem. We have generalized the case where all the data are given at one point to vector data [6] and even matrix data [17]. In [18] we allowed data to be given at two points (zero and infinity). In this paper, we generalize this solution method to the case of vector interpolation data with possible non-coinciding interpolation points.

The essential observations made in the above mentioned papers is that the RIP can be linearized whereupon it reduces to the recursive solution of nested systems of homogeneous linear equations. The matrices of these systems have a special structure. Thus we have to find an element from their null space. Since these systems have one more column than they have rows, their null space is at least one dimensional. In the case of Padé approximation, these systems are (block) Toeplitz or (block) Hankel. These matrices appear in many different seemingly unrelated mathematical topics and engineering applications, which might explain their popularity during the last decades. You will find them for example in the theory and applications of orthogonal polynomials, integral equations, digital filtering and signal processing, time series analysis, stochastic processes, inverse scattering, BCH-decoding etc., to name just a few.

The basic algorithms to solve these problems are the Levinson/Schur/Szegő algorithms (for the Toeplitz case) and the Euclidean/Berlekamp-Massey/Kronecker algorithms (for the Hankel case). A basic element in these algorithms, and also in our approach, is the description of all elements in the null space of matrices with a special structure. This is done by constructing a basis. The nesting of the matrices and their special structure make it possible to find such a basis rather efficiently if you know the bases of the previous null spaces. During the last decade much attention has been paid to a multitude of generalizations of these algorithms to all sorts of "special" or "structured" matrices. People have considered matrices close-to-Toeplitz or Hankel, Hilbert matrices and Vandermonde matrices [7] and many others.

Recently, there is a revived interest in the rational interpolation case [15, 14, 16, 2, 3, 1, 19]. The matrices involved here are Löwner matrices or generalized Löwner matrices depending on whether the interpolation points can coincide or not (see [2]). The basic thing to be studied remains, just like in the classical problems, to describe a basis for the null spaces of such matrices and to find an efficient way to generate them. In this paper, you will find a solution to this problem.

Vector rational interpolants have been considered before. The papers by Peter Graves-Morris are the most recent ones. [10, 11, 13]. We refer to his introduction (especially the second one) to find out about the history of the problem and the relation to related problems in the literature like the so called Latin and German polynomial problems. Our algorithm will be able to recover the directional vector interpolants considered in [11] (see section 2).

In this paper we combine the new approach to rational interpolation of [19], and the idea of minimal vector Padé approximation of [6] to the case where vector interpolation data are given. All the results we obtained for the scalar problem are generalized to the vector case. The basic results being the constructive way of defining the basis of the null space, which results in an efficient recursive algorithm (section 3). A "minimal" solution of the original interpolation problem (section 2) can then be readily given (section 4) as well as a parameterization of all solutions, not only for the linearized problem, but also for the proper interpolation problem (section 5).

The absence in our algorithm of a reordering strategy which is needed to overcome a singular situation when the algorithm hits a "block" in the Newton-Padé table in most of the existing scalar methods [8, 12, 9, 22, 23, 24, 25] is a basic advantage. For the vector case, a normal situation was supposed from the start [10, 11, 13]. Our algorithm will also work in singular situations for the vector case.

2 The vector rational interpolation problem ($\underline{\text{V}}$RIP)

This paper is a continuation of [19] and [6]. Ample motivation for our approach and the definitions and methods introduced can be found there. It might be an advantage to read the simpler cases of scalar interpolants [19], or completely confluent points [6] first. The present paper follows closely the outline of [19].

The field, finite or infinite, we are working with , is denoted by $\mathbb{K}$. The set of polynomials with coefficients of the field $\mathbb{K}$, is denoted by $\mathbb{K}[z]$. The set of polynomial m-vectors, i.e., the set of all vectors whose m elements belong to $\mathbb{K}[z]$, is denoted by $\mathbb{K}^m[z]$. Suppose we want to interpolate the given data $f_i \in \mathbb{K}^m$, in the interpolation points $z_i \in \mathbb{K}, i = 1, 2, \ldots, p$, by a vector rational function $n(z)/d(z)$ with $n(z) \in \mathbb{K}^m[z]$ and $d(z) \in \mathbb{K}[z]$. The division by $d(z)$ is taken componentwise, i.e.

$$n(z)/d(z) = \begin{bmatrix} n_1(z)/d(z) \\ n_2(z)/d(z) \\ \vdots \\ n_m(z)/d(z) \end{bmatrix}.$$

Because we want to keep the exposition here as simple as possible, we don't allow a component of f_i to have the value ∞. Hence, $d(z_i) \neq 0, i = 1, 2, \ldots, p$. For the same simplicity reasons, we also do not allow an interpolation point at infinity ($z_i \neq \infty, i = 1, 2, \ldots, p$). These restrictions are basically made to simplify the notation and to avoid a lot of exceptional cases. However, the same method still works. The extensions for the scalar case are covered in [20]. See also [15]. For the same reason of ease of exposition, we shall initially suppose that all the interpolation points are different ($z_i \neq z_j, i \neq j$). The confluent case is postponed until section 6.

Every polynomial $(m+1)$-tuple $(n(z), d(z)) = (n_1(z), n_2(z), \ldots, n_m(z), d(z))$ which solves the proper interpolation conditions

$$f_i = n(z_i)/d(z_i), \quad i = 1, 2, \ldots, p \tag{1}$$

shall also satisfy the linearized conditions

$$f_i d(z_i) = n(z_i), \quad i = 1, 2, \ldots, p. \tag{2}$$

However, the converse is only guaranteed if $d(z_i) \neq 0, i = 1, 2, \ldots, p$. Note that we don't consider

$$\frac{n(z)p(z)}{d(z)p(z)} \quad \text{with} \quad p(z) \in \mathbb{K}[z] \setminus \{0\}$$

as a solution satisfying the proper interpolation conditions as soon as $p(z_i) = 0$ for some interpolation point $z_i, i = 1, 2, \ldots, p$, even if $n(z)/d(z)$ satisfies the proper interpolation conditions with $d(z_i) \neq 0, i = 1, 2, \ldots, p$. Because by assumption none of the components of f_i is ∞, $d(z_i) = 0$ implies $n(z_i) = 0$. In other words, $n(z_i) \neq 0$ implies $d(z_i) \neq 0$. This means that if any of the components of the polynomial m-tuple $n(z)$ is nonzero in z_i then $d(z_i) \neq 0$. Thus, for a solution of (2), $d(z_i) \neq 0$, under the given restrictions, is equivalent with $(n(z_i), d(z_i)) \neq 0$. This can be expressed by saying that $(n(z), d(z))$ is not divisible by $z - z_i$. Therefore we can reformulate the proper problem (1) as the linearized problem (2) with the extra condition that $(n(z), d(z))$ is not divisible by $z - z_i, i = 1, 2, \ldots, p$. This latter condition is simpler to work with in the proofs.

Example 1 Suppose we are given the following interpolation data

$$z_1 = 0, \quad f_1 = [1\ 0]^T, \quad z_2 = 1, \quad f_2 = [2\ 1]^T, \quad z_3 = 2, \quad f_3 = [2\ 1]^T.$$

Of course, there are many solutions of the proper problem, for example

$$n_1(z) = 1 + \frac{3}{2}z - \frac{1}{2}z^2, \quad n_2(z) = \frac{3}{2}z - \frac{1}{2}z^2, \quad d(z) = 1 \tag{3}$$

is a solution, but also

$$n_1(z) = 1 + \frac{7}{2}z - \frac{1}{2}z^2, \quad n_2(z) = \frac{5}{2}z - \frac{1}{2}z^2, \quad d(z) = 1 + z \tag{4}$$

and

$$n_1(z) = 1 + \frac{7}{2}z - \frac{1}{2}z^2, \quad n_2(z) = \frac{9}{2}z - \frac{7}{2}z^2 + z^3, \quad d(z) = 1 + z \tag{5}$$

are solutions. On the other hand

$$n_1(z) = 2z, \quad n_2(z) = \frac{1}{2}z^2(3 - z), \quad d(z) = z \tag{6}$$

satisfies the linearized conditions (2) but not the proper conditions (1). We note that indeed $(n(z), d(z))$ is divisible by $z - z_1 = z$.

As we have seen in this example, there are many possible solutions to a set of interpolation data. The set of solutions is never empty since we can always choose $d(z) = 1$ and for $n(z)$ the interpolating polynomials. Similar to the scalar case [19] and the vector Padé case [6], we want to select the simplest possible solution in a well defined sense. We could for example minimize the maximal degree of the components of the polynomial $(m+1)$-couple $(n(z), d(z))$. Thus choose the solution for which

$$\max\{\deg n_1(z), \deg n_2(z), \ldots, \deg n_m(z), \deg d(z)\}$$

is minimal (we define $\deg 0 = -\infty$). This will not completely order the solutions, since in example 1 (3) and (4) are different solutions with the same maximal degree. In any case, such a criterion will tend to distribute the degrees of freedom as uniformly as possible over all the components of $(n(z), d(z))$, i.e. make the degrees of all these components nearly equal. For $m = 1$, the scalar case, this corresponds to solutions near the main diagonal of the Newton-Padé interpolation table. In order to get other entries of the interpolation table as minimal solutions, we can play with an integer shift parameter s and require $\max\{\deg n(z) - s, \deg d(z)\}$ to be minimal. Analogously, we define for the vector case the shift parameters $s = (s_1, s_2, \ldots, s_m) \in \mathbb{Z}^m$ and minimize

$$\alpha = \max(\deg n_1(z) - s_1, \deg n_2(z) - s_2, \ldots, \deg n_m(z) - s_m, \deg d(z)\}.$$

α is called the s-degree of the $(m+1)$-tuple $(n(z), d(z))$. When all $s_i = 0, i = 1, 2, \ldots, m$, then we are back in the "main" diagonal case we proposed first. For a first reading of this paper, it might be simplest to think of the s_i as being zero. The shift parameters define the structure of the polynomial couple $(n(z), d(z))$. More precisely, $(n(z), d(z))$ has s-degree $\leq \alpha$ iff $\deg n_j(z) \leq \alpha + s_j, j = 1, \ldots, m$ and $\deg d(z) \leq \alpha$. Given the shift parameters $s \in \mathbb{Z}^m$, we take the s-degree as the complexity of the polynomial $(m+1)$-tuple $(n(z), d(z))$. We can now define the following two vector rational interpolation problems (V̲RIPs).

Definition 1 (Linearized Vector Rational Interpolation Problem (L̲VRIP)) *The interpolation points $z_i \in \mathbb{K}$ and data $f_i \in \mathbb{K}^m, i = 1, 2, \ldots, p$ and the shift parameters $s \in \mathbb{Z}^m$ are given. We look for a polynomial $(m+1)$-tuple $(n(z), d(z))$ of minimal s-degree, satisfying the linearized rational interpolation conditions*

$$f_i d(z_i) = n(z_i), i = 1, 2, \ldots, p. \tag{7}$$

A solution of the L̲VRIP could be divisible by $z - z_i$ for a certain $i \in \{1, 2, \ldots, p\}$. Therefore, we consider also the following problem.

Definition 2 (Proper Vector Rational Interpolation Problem (P̲VRIP)) *This is the linearized rational interpolation problem with the additional condition that $(n(z), d(z))$ is not divisible by $z - z_i, i = 1, 2, \ldots, p$.*

To illustrate the previous notions and concepts, we introduce the following example.

Example 1 (continued) The interpolation data of example 1 are given. Suppose we want to derive a vector rational interpolant $[n_1(z)\ n_2(z)]^T/d(z)$ with the following degree-structure. We allow the degree of $n_2(z)$ to be one higher than the degrees of $n_1(z)$ and $d(z)$. In other words, we look for a rational interpolant with $\deg n_1(z) \leq \alpha$, $\deg n_2(z) \leq \alpha + 1$ and $\deg d(z) \leq \alpha$. It is clear that without limiting the value of α we get an infinity of solutions, not only for the L̲VRIP but also for the P̲VRIP where all these solutions have the prescribed degree structure. Therefore, from all these solutions, we are going to take those with α as small as possible. Thus we look for those solutions with smallest s-degree with $s = (0, 1)$.

We give now the relation between our L̲VRIP and the directional vector-valued interpolants (**DIRI**s) of [11]. For the **DIRI** problem, we also have to consider $N+1$ interpolation points z_i, $i = 0, 1, \ldots, N$ with corresponding function values $f(z_i) = v_i \in \mathbf{C}^m$. Besides these, there are given k_l independent directions $w_k \in \mathbf{C}^m$, $k = 1, \ldots, k_l$ and associated multiplicities m_k, such that $\sum_{k=1}^{k_l} m_k = M \leq N$. Along the direction w_k, the numerator degree should be reduced to $N - m_k$ while the denominator degree can increase up to M. This means the following. Let $P(z)$ be the polynomial m-tuple, representing the numerator of the **DIRI** and let $Q(z)$ be the corresponding denominator. Then $P(z) \cdot w_k$ should have a degree at most $N - m_k$, where $a \cdot b$ denotes the usual Euclidean inner product of the two m-vectors a and b. This condition we can rewrite as follows. Let $W \in \mathbf{C}^{m \times m}$ be a matrix whose first k_l columns are the directional vectors w_k and the remaining columns are added to make W a regular square matrix. Then

$$\text{degree } [P_1(z), \ldots, P_m(z)]W \leq [N - m_1, \ldots, N - m_{k_l}, N, \ldots, N].$$

The trailing $m - k_l$ bounds impose no extra conditions so that we can consider them as redundant. We only need them as a trick to define the invertible matrix W. Now the interpolation conditions $P(z_i)/Q(z_i) = v_i$ can be rewritten as the equivalent conditions (consider $P(z)$ and v_i as row-vectors) $P(z_i)W/Q(z_i) = v_i W$, $i = 0, 1, \ldots, N$. Rename $N+1$ as p, $P(z)W$ as $n(z)$, $Q(z)$ as $d(z)$ and $v_i W$ as f_{i+1} and you will recognize our L̲VRIP with shift parameters $s_j = N - M - m_j$, $M = \sum_1^{k_l} m_k$. Thus it turns out that one of the solutions that our algorithm will produce with appropriate shift parameters, will be a **DIRI**.

In [11], the link between the German polynomial problem and the construction of **DIRI**s is explained. Thus our method is also able to find solutions of the German polynomial problem.

3 The solution of the L̲VRIP

In this and the subsequent sections we shall work constantly with polynomial $(m+1)$-tuples. These will be split into an m-tuple of numerator polynomials and one scalar denominator polynomial. If the $(m+1)$-tuple is denoted by e.g., $v(z)$, then the corresponding numerator m-tuple is denoted by $n_v(z)$ and the associated denominator polynomial as $d_v(z)$. Thus $v(z) = (n_v(z), d_v(z))$. It will happen that we have a sequence of polynomial $(m+1)$-tuples : $v_j(z)$. In order not to complicate the notation, we shall indicate the numerator and denominator also with that index : $v_j(z) = (n_j(z), d_j(z))$ instead of $(n_{v_j}(z), d_{v_j}(z))$.

We can write (7) as a set of linear homogeneous equations:

$$R_i v(z) = 0, i = 1, 2, \ldots, p \tag{8}$$

where $R_i v(z)$ is the ith residual of the polynomial $(m+1)$-tuple $v(z) = (n_v(z), d_v(z))$ defined as $R_i v(z) = f_i d_v(z_i) - n_v(z_i)$. We denote by $S_{p,\alpha}$ the set of all polynomial $(m+1)$-tuples $(n(z), d(z))$ having s-degree $\leq \alpha$ and satisfying (7) or, what is the same, solving (8).

Theorem 1 *The set $S_{p,\alpha}$ forms a vector space over the field $\mathbb{K}$.*

Proof. Take $v(z), w(z) \in S_{p,\alpha}$, i.e. $R_i v(z) = 0$ and $R_i w(z) = 0, i = 1, 2, \ldots, p$. Then, also $av(z) + bw(z) \in S_{p,\alpha}$ with $a, b \in \mathbb{K}$ because $R_i(av(z) + bw(z)) = aR_i v(z) + bR_i w(z) = 0$. □

An $(m+1)$-tuple $v(z)$ which solves (8) is said to be a solution of level p. We shall make this explicit by writing $v_p(z)$ instead of $v(z)$. Note that the $(m+1)$-tuple $v_p(z)$ of level p shall satisfy (8) and hence gives $R_i v_p(z) = 0$ for $i = 1, 2, \ldots, p$, but if there are more interpolation points $z_i, i = p+1, p+2, \ldots$, then the residuals need not be zero anymore. Thus, in general, $R_i v_p(z) \neq 0$ for $i > p$. We show next that a basis $BS_{p,\alpha}$ for the vector space $S_{p,\alpha}, -\infty < \alpha < +\infty$, can be constructed from $(m+1)$ polynomial $(m+1)$-tuples $v_p^j(z) = (n_p^j(z), d_p^j(z)),\ j = 1, 2, \ldots, m+1$, of level p. The $m+1$ polynomial $(m+1)$-tuples of level p are numbered with a superscript j. Thus $v_p^j(z)$ is the jth $(m+1)$-tuple of level p. Without loss of generality, we can assume that the $v_p^j(z)$ are ordered with respect to increasing s-degree. The ordering of the $v_p^j(z)$ having equal s-degree is arbitrary. The ordering is not essential but it simplifies the notation. In our proof, the residuals $R_i v_p^j(z)$ for $i > p$ will be important, because if $R_{p+1} v_p^j(z) = 0$, then $v_p^j(z)$ is not only a solution for level p but also for level $p+1$. Therefore, we introduce the notion of residual space $R_{i,p}^j$ for all $i > 0$ and $1 \leq j \leq m+1$ as

$$R_{i,p}^j = \operatorname{span}_{k=1}^{j} R_i v_p^k(z).$$

In other words, $R_{i,p}^j$ is the vector space spanned by the ith residuals (those at z_i) of the first j (the ones with smallest s-degree) $(m+1)$-tuples $v_p^k(z)$ of level p. Of course, for $i \leq p$ this space contains just the zero vector. However, it will turn out that for $i > p$, the residuals of all the $(m+1)$-tuples $v_p^j(z), j = 1, 2, \ldots, m+1$ will span the whole space $\mathbb{K}^m$, i.e. $R_{i,p}^{m+1} = \mathbb{K}^m$ for $i > p$.

The basis $BS_{p,\alpha}$ will in general have a dimension larger than $m+1$. Thus the $(m+1)$ solutions $v_p^j(z), j = 1, \ldots, m+1$ will not be sufficient to give the whole basis. We shall also need shifted versions of them. Therefore, we introduce the following short hand notation. Given an integer number α and a polynomial $(m+1)$-tuple $x(z)$ having s-degree α_x, we define the set of polynomial $(m+1)$-tuples $\{x(z)\}^\alpha$ as follows:

$$\begin{aligned} \{x(z)\}^\alpha &= \varnothing \quad , \quad \text{if } \alpha < \alpha_x \\ \{x(z)\}^\alpha &= \{x(z), zx(z), \ldots, z^{\alpha-\alpha_x} x(z)\} \quad , \quad \text{if } \alpha \geq \alpha_x. \end{aligned}$$

In other words, the set $\{x(z)\}^\alpha$ is empty if $\alpha < \alpha_x$, and if $\alpha \geq \alpha_x$ it contains the polynomial $(m+1)$-tuple $x(z)$ and shifted versions of $x(z)$ having s-degrees ranging from α to α_x. We shall prove the following theorem:

Theorem 2 *For each $p \geq 0$, there exist $(m+1)$ polynomial $(m+1)$-tuples $v_p^j(z), j = 1, 2, \ldots, m+1$ (from level p) such that for each $\alpha, |\alpha| \leq \infty$, a basis $BS_{p,\alpha}$ for $S_{p,\alpha}$, is given by*

$$BS_{p,\alpha} = \cup_{j=1}^{m+1} \{v_p^j(z)\}^\alpha.$$

The residual space $R_{i,p}^{m+1}$ is equal to $\mathbb{K}^m$ for $i > p$.

Note that the basis $BS_{p,\alpha}$ contains all the polynomial $(m+1)$-tuples $v_p^j(z)$ of level p with s-degree at most α. If some $v_p^j(z)$ has an s-degree less than α, we add also its shifted versions $z^k v_p^j(z)$ that have an s-degree not raising above α.

Proof. We shall prove the theorem by induction on p. First we prove that the theorem is true for $p = 0$ if we choose e.g., $v_0^j(z) = e_j$, $j = 1, 2, \ldots, m+1$, with e_j the $(m+1)$-tuple with all the components equal to zero except the jth one which is equal to one. At level 0, there are no interpolation conditions. Therefore, the elements of $S_{0,\alpha}$ are all polynomial $(m+1)$-tuples of s-degree $\leq \alpha$, i.e. they can be parametrized as $(a_0^1 + a_1^1 z + \cdots + a_{\alpha+s_1}^1 z^{\alpha+s_1}, \ldots, a_0^{m+1} + a_1^{m+1} z + \cdots + a_{\alpha+s_{m+1}} z^{\alpha+s_{m+1}})$ with $s_{m+1} = 0$. The $(m+1)$-tuple $v_0^j(z)$ has s-degree $-s_j$. Hence, $v_0^j(z)$ can be shifted $\alpha - (-s_j) = \alpha + s_j$ times before its s-degree is becoming greater than α. The residual space $R_{i,0}^{m+1} = \text{span}_{k=1}^{m+1} R_i v_0^k(z) = \text{span}\{-e_1', -e_2', \ldots, -e_m', f_i\} = \mathbb{K}^m$ for $i > 0$ with e_i' the ith identity vector $\in \mathbb{K}^m$. Note that the v_0^j are not necessarily ordered by increasing s-degree, but knowing that v_0^j has s-degree $-s_j$, this ordering can be easily obtained if this were desired.

To prove the induction step, we assume now that the theorem is true for a certain value for p. Without loss of generality we can assume that the polynomial $(m+1)$-tuples $v_p^j(z)$ are ordered such that the s-degrees α_p^j of $v_p^j(z)$ satisfy $\alpha_p^j \leq \alpha_p^{j+1}$ for $j = 1, 2, \ldots, m$.

The proof is constructive because an algorithm is given to compute a possible choice for the set of all the polynomial $(m+1)$-tuples $v_{p+1}^j(z)$ for level $p+1$, given all the $v_p^j(z)$ of the previous level. First of all we shall describe how to compute a nonzero polynomial $(m+1)$-tuple $a_{p+1}(z) \in S_{p+1,\alpha}$, not divisible by $z - z_{p+1}$ (which always exists) having minimal s-degree α. This polynomial $(m+1)$-tuple $a_{p+1}(z)$ together with m other polynomial $(m+1)$-tuples, which are then readily found, will give us $v_{p+1}^j(z), j = 1, 2, \ldots, m+1$ (up to reordering). Because $a_{p+1}(z)$ has to be in $S_{p+1,\alpha} \subset S_{p,\alpha}$ (one interpolation condition more), we can write $a_{p+1}(z)$ as a linear polynomial combination of the polynomial $(m+1)$-tuples $v_p^j(z)$. Because $S_{p,\alpha} = \{(0, 0, \ldots, 0)\}$ for $\alpha < \alpha_p^1$ ($v_p^1(z)$ has smallest possible s-degree at level p), the problem is trivial for $\alpha < \alpha_p^1$. Therefore, suppose $\alpha \geq \alpha_p^1$. By induction hypothesis, we know that $R_{p+1,p}^{m+1} = \mathbb{K}^m$. Consequently, m of the $(m+1)$ residuals $R_{p+1} v_p^j(z), j = 1, 2, \ldots, m+1$, are linearly independent. We show that $a_{p+1}(z)$ can not be a linear polynomial combination of only those $v_p^j(z)$ having linearly independent residuals. Suppose $a_{p+1}(z)$ is such a linear polynomial combination:

$$a_{p+1}(z) = \sum_{j \in J} p_j(z) v_p^j(z)$$

with $J \neq \varnothing$ and all the residuals $R_{p+1} v_p^j, j \in J$ linearly independent. Because $a_{p+1}(z)$ has to satisfy the $(p+1)$th interpolation condition, the residual $R_{p+1} a_{p+1}(z)$ has to be equal to the zero m-vector, i.e.,

$$R_{p+1} a_{p+1}(z) = \sum_{j \in J} p_j(z_{p+1}) R_{p+1} v_p^j(z) = 0.$$

Because the residuals in play are linearly independent, this means that $p_j(z_{p+1}) = 0$ for $j \in J$. Hence, $z - z_{p+1}$ divides $a_{p+1}(z)$. Therefore, to compute a possible choice for $a_{p+1}(z)$, we have to consider a linear polynomial combination of $v_p^j(z)$ where the corresponding residuals are linearly dependent. To keep the s-degree α as small as possible, we compute the smallest j for which $R_{p+1} v_p^i(z), i = 1, 2, \ldots, j$ are linearly dependent, thus the smallest j such that $\dim R_{p+1,p}^j < j$. Then we know that the residual corresponding to $v_p^j(z)$ can be written as

$$R_{p+1} v_p^j(z) = \sum_{k=1}^{j-1} c_k R_{p+1} v_p^k(z), \quad \text{or} \quad R_{p+1}(v_p^j(z) - \sum_{k=1}^{j-1} c_k v_p^k(z)) = 0.$$

Therefore, $v_p^j - \sum_{k=1}^{j-1} c_k v_p^k$ is a possible choice for $a_{p+1}(z)$ if we can prove that $a_{p+1}(z)$ is then not divisible by $z - z_{p+1}$. If $a_{p+1}(z)$ would be divisible by $z - z_{p+1}$, then $a'(z) = a_{p+1}(z)/(z - z_{p+1}) \in S_{p,\alpha-1}$ since indeed, the s-degree is decreased by 1 and all the interpolation conditions, except possibly the one for z_{p+1} are still satisfied. If $a'(z) \in S_{p,\alpha-1}$, then it can be written in a unique way as a linear polynomial combination of the $v_p^k(z)$ having s-degree less than α. Hence, $v_p^j(z)$ is not involved. Because $a_{p+1}(z) = (z - z_{p+1})a'(z)$, $a_{p+1}(z)$ can be written as a unique linear polynomial combination not involving $v_p^j(z)$. However, this is in contradiction to the fact that $v_p^j(z)$ is a component of $a_{p+1}(z)$ since there is only one way to write $a_{p+1}(z) \in S_{p,\alpha}$ in terms of the basis vectors $\cup_{j=1}^{m+1}\{v_p^j(z)\}^\alpha$.

We shall prove now that $a_{p+1}(z)$ together with the m polynomial $(m+1)$-tuples $(z - z_{p+1})v_p^k(z), k \neq j$ are a possible choice for $v_{p+1}^j(z), j = 1, 2, \ldots, m+1$ by proving that for each α a basis for $S_{p+1,\alpha}$ is given by $BS_{p+1,\alpha} = \cup_{k\neq j}\{(z - z_{p+1})v_p^k(z)\}^\alpha \cup \{a_{p+1}(z)\}^\alpha$. Take $a(z) \in S_{p+1,\alpha}$, then we first show that $a(z)$ can be written in a unique way as $a(z) = ca_{p+1}(z) + a'(z)$ with $a'(z)$ divisible by $z - z_{p+1}$ and $c \in \mathbb{K}$ constant. If $a(z)$ has s-degree smaller than the s-degree of $a_{p+1}(z)$, then $a(z)$ is divisible by $z - z_{p+1}$ since by construction $a_{p+1}(z)$ is a solution of level $p+1$, not divisible by $z - z_{p+1}$ with the smallest possible s-degree. So, if $a(z) \in S_{p+1,\alpha}$, it can only have an s-degree smaller than the s-degree of $a_{p+1}(z)$ if it is divisible by $z - z_{p+1}$. Hence, in this case $c = 0$. When the s-degree of $a(z)$ is greater than or equal to the s-degree of $a_{p+1}(z)$, we can always choose a unique $c \in \mathbb{K}$ such that $a(z) - ca_{p+1}(z) = (n_a(z), d_a(z)) - c(n_{p+1}(z), d_{p+1}(z))$ is divisible by $z - z_{p+1}$, i.e. such that

$$d_a(z_{p+1}) - cd_{p+1}(z_{p+1}) = 0 \in \mathbb{K} \tag{9}$$

and

$$n_a(z_{p+1}) - cn_{p+1}(z_{p+1}) = 0 \in \mathbb{K}^m. \tag{10}$$

Because $a_{p+1}(z)$ is not divisible by $z - z_{p+1}$, either the scalar $d_{p+1}(z_{p+1}) \neq 0$ or the m-vector $n_{p+1}(z_{p+1}) \neq 0$. Suppose $d_{p+1}(z_{p+1}) \neq 0$, the other case is similar. We can determine c as $c = d_a(z_{p+1})/d_{p+1}(z_{p+1})$. Because $R_{p+1}a(z)$ and $R_{p+1}a_{p+1}(z)$ are equal, $R_{p+1}a(z) = R_{p+1}a_{p+1}(z) = 0$ and we obtain:

$$f_{p+1}(d_a(z_{p+1}) - cd_{p+1}(z_{p+1})) = n_a(z_{p+1}) - cn_{p+1}(z_{p+1}). \tag{11}$$

Assuming (9) is true, (10) follows easily from (11). Thus in all cases, c in the decomposition $a(z) = ca_{p+1}(z) + a'(z)$ is uniquely defined. Hence, also $a'(z)$ is uniquely defined. Now we prove that $a'(z)$ can be written as a combination of the remaining basis vectors $\cup_{k\neq j}\{(z - z_{p+1})v_p^k(z)\}^\alpha$. Since $a'(z) = (z - z_{p+1})a''(z)$ with $a''(z) \in S_{p,\alpha-1}$, we can use the induction hypothesis to write $a''(z)$ as a combination of the basis $\cup_{k\neq j}\{v_p^k(z)\}^{\alpha-1}$, or equivalently, $a'(z)$ can be written as a combination of $\cup_{k\neq j}\{(z - z_{p+1})v_p^k(z)\}^\alpha$. Thus we have proved that

$$\cup_{j=1}^{m+1}\{v_{p+1}^j(z)\}^\alpha = \cup_{k\neq j}\{(z - z_{p+1})v_p^k(z)\}^\alpha \cup \{a_{p+1}(z)\}^\alpha$$

forms a basis for $S_{p+1,\alpha}$.

It remains to be shown that $R_{i,p+1}^{m+1} = \mathbb{K}^m, i > p+1$. By definition, we have that

$$\begin{aligned} R_{i,p+1}^{m+1} &= \text{span}_{\,k\neq j}\{R_i(z - z_{p+1})v_p^k(z), R_i(v_p^j - \sum_{l=1}^{j-1} c_l v_p^l)\} \\ &= \text{span}_{\,k\neq j}\{(z_i - z_{p+1})R_i v_p^k(z), R_i v_p^j - \sum_{l=1}^{j-1} c_l R_i v_p^l\}. \end{aligned}$$

Because $z_i \neq z_{p+1}$ and $R_{i,p}^{m+1} = \mathbb{K}^m$ for $i > p+1$, it is clear that $R_{i,p+1}^{m+1} = \mathbb{K}^m$, $i > p+1$ and this proves the theorem completely. $\square$

The proof of the theorem contains an algorithm to compute the $(m+1)$ polynomial $(m+1)$-tuples $v_p^j(z), j = 1, 2, \ldots, m+1$ at each level p knowing the $v_{p-1}^j(z)$ of the previous level. The first one $v_p^1(z)$, will have a minimal s-degree and it will be a solution for the LVRIP. If $\alpha_p^1 = \alpha_p^2 = \cdots = \alpha_p^l < \alpha_p^{l+1}$ (with $\alpha_p^{m+1} = +\infty$), it is clear that also $v_p^2(z), v_p^3(z), \ldots, v_p^l(z)$ are solutions of the LVRIP. These will allow us to give a parameterization of all solutions. We refer to theorem 4. To be able to give a compact formulation of the algorithm, we introduce the following notations:

- We group the $(m+1)$ polynomial $(m+1)$-tuples $v_p^j(z)$ together in an $(m+1)\times(m+1)$ polynomial matrix as follows

$$\begin{aligned} G_p(z) &= \left[\begin{array}{cccc} v_p^1(z) & v_p^2(z) & \cdots & v_p^{m+1}(z) \end{array}\right] \\ &= \left[\begin{array}{cccc} n_p^1(z) & n_p^2(z) & \cdots & n_p^{m+1}(z) \\ d_p^1(z) & d_p^2(z) & \cdots & d_p^{m+1}(z) \end{array}\right]. \end{aligned}$$

- The elementary polynomial operations, which we shall apply on $(m+1)\times(m+1)$ polynomial matrices, are denoted as follows:
  - $E_{i,j}$ denotes the permutation matrix describing a switching of the columns i and j,
  - $E_i(p(z))$ denotes the polynomial matrix describing the multiplication of column i by a polynomial $p(z) \in \mathbb{K}[z]$,
  - $E_{i,j}(c)$ denotes the elementary matrix describing the addition to column i of column j multiplied by the scalar $c \in \mathbb{K}$.
- U denotes the $(m+1)\times(m+1)$ identity matrix.

We can summarize the algorithm as in algorithm VRIP below.

It is clear that we can write:

$$G_p(z) = V_0(z)V_1(z)\cdots V_p(z) \text{ with}$$

$$\begin{aligned} V_0(z) &= \left[\begin{array}{cc} x_0(z) & t_0(z) \\ y_0(z) & u_0(z) \end{array}\right] \\ &= P_0 \end{aligned}$$

and for $i \geq 1$

$$\begin{aligned} V_i(z) &= \left[\begin{array}{cc} x_i(z) & t_i(z) \\ y_i(z) & u_i(z) \end{array}\right] \\ &= \prod_{k=1}^{j_i-1} E_{j_i,k}(-c_k^i) \prod_{k\neq j_i} E_k(z-z_i), \end{aligned} \tag{12}$$

where j_i and c_k^i are the j and c_k values needed in the algorithm to go from level $i-1$ to level i. The partitioning of the $V_i(z)$ matrices into $x_i(z), y_i(z), t_i(z)$ and $u_i(z)$ is such that $y_i(z)$ is a scalar polynomial concentrating our attention on $v_i^1(z)$. These $V_i(z)$ matrices can

Algorithm 1: VRIP{vector rational interpolation}

{ Initialization }
G_0 is initialized as $G_0(z) = UP_0 = V_0(z)$,
where P_0 is a sequence of elementary operations
of the form $E_{i,j}$, i.e. describing a permutation matrix,
ordering the v_0^j with respect to increasing s-degree.
for $p = 0, 1, 2, \ldots$
  Look for the smallest j such that $R_{p+1}v_p^j(z)$
  can be written as a linear combination of the previous residuals:
  $R_{p+1}v_p^j(z) = \sum_{k=1}^{j-1} c_k R_{p+1} v_p^k(z)$.
  If we set $V'_{p+1}(z) = \prod_{k=1}^{j-1} E_{j,k}(-c_k) \prod_{k\neq j} E_k(z - z_{p+1})$,
  then we get $G_{p+1}(z)$ by permutation of $G_p(z)V'_{p+1}(z)$
  $G_{p+1}(z) = G_p(z)V_{p+1}(z)$ with
  $V_{p+1}(z) = V'_{p+1}(z)P_{p+1}$
  where P_{p+1} is a permutation matrix reordering the polynomial
  vectors with respect to their s-degree.
endfor

be seen as generalized Möbius transformations leading to a generalized continued fraction forking not only downwards but also upwards, (see [6]). This continued fraction has not only scalar but also vector and matrix entities involved but all the divisions are by scalars. It follows that $n_p^1(z)/d_p^1(z)$, a solution of the $\underline{\text{L}}$VRIP, is the pth convergent of the generalized continued fraction:

$$\frac{x_0(z) + t_0(z)\dfrac{x_1(z) + t_1(z)\frac{\cdots}{\cdots}}{y_1(z) + u_1(z)\frac{\cdots}{\cdots}}}{y_0(z) + u_0(z)\dfrac{x_1(z) + t_1(z)\frac{\cdots}{\cdots}}{y_1(z) + u_1(z)\frac{\cdots}{\cdots}}}. \tag{13}$$

Note that we can also write a similar continued fraction whose pth convergent is $n_p^j(z)/d_p^j(z)$, $j > 1$ by partitioning the $V_i(z)$ matrices in another way. Remember that the polynomial $(m+1)$-tuple $a_{p+1}(z)$ which featured in the proof of the theorem is one of the $v_{p+1}^j(z)$. More precisely, it is the solution of the $\underline{\text{L}}$VRIP with the additional condition that this polynomial tuple is not divisible by $z - z_{p+1}$. The above continued fraction reduces to an ordinary continued fraction, at least formally, if all the $t_i(z)$ vanish. It still would have vector valued convergents. Since the multiplications with the $V_i(z)$ matrices represent more general transformations, we get the more general form of the fractions. It is however possible to get ordinary continued fractions as well. See e.g., [5, Theorem 2.9].

The algorithm, described above, can be implemented in several ways. The linearized rational interpolation conditions (7) can be written in terms of the classical basis $1, z, z^2, \ldots$, involving classical block-Vandermonde matrices. In this case, the multiplication of a polynomial by $z - z_{p+1}$, needed in the algorithm, is straightforward. Another possibility is to write the interpolation conditions (7) in terms of a basis of orthogonal polynomials $\phi_0(z), \phi_1(z), \phi_2(z), \ldots$, involving generalized block-Vandermonde matrices. The multiplication by $z - z_{p+1}$ of a polynomial written in terms of this basis of orthogonal polynomials can be carried out using the recurrence relation for orthogonal polynomials: $\phi_k(z) = \lambda_k(z - \alpha_k)\phi_{k-1}(z) - \beta_k\phi_{k-2}(z)$. More generally, we can use any basis as long as the shift operation, i.e., the multiplication by z of a polynomial written in terms of this basis, can be carried out in a simple and efficient way.

If we consider $G_p(z) = [v_p^1(z)\ v_p^2(z)\ \cdots\ v_p^{m+1}(z)] = V_0(z)V_1(z)\cdots V_p(z)$ as an abstract layered medium, with layers described by $V_i(z)$, each step of the algorithm, given above, computes the residuals $R_{p+1}v_p^j(z)$, needed to compute the next layer $V_{p+1}(z)$, and connects this new layer to the layered medium by computing $G_{p+1}(z) = G_p(z)V_{p+1}(z)$. So, in the terminology of Bruckstein and Kailath [4], we can call this a layer adjoining algorithm. However, it is not necessary to obtain the layered medium $G_p(z)$ explicitly. If we know the separate layers $V_i(z)$, we can represent the layered medium by the continued fraction (13). To compute the layer $V_p(z)$, we need to know the residuals $R_p v_{p-1}^j(z)$. Suppose we know all the residuals $R_i v_{p-1}^j(z)$ of level $p-1$, with $i \geq p$. It is easy to design an algorithm computing the next layer $V_p(z)$ and transforming all the residuals of the previous step into the residuals of level p, using the formula

$$\left[R_i v_p^1(z)\ R_i v_p^2(z)\cdots R_i v_p^{m+1}(z)\right] = \left[R_i v_{p-1}^1(z)\ R_i v_{p-1}^2(z)\cdots R_i v_{p-1}^{m+1}(z)\right] V_p(z_i).$$

This algorithm is called a layer peeling one [4], because at each step p we peel off the layer $V_p(z)$. Compared to the layer adjoining algorithm, the layer peeling one has the drawback that we have to know all the residuals $R_i v_0^j(z)$, with $i \geq 1$ at the beginning of the algorithm. As an advantage, we mention that the layer peeling algorithm is easier to parallelize than the other one. From the previous explanation, it should be clear that also a mixture of a layer adjoining and layer peeling algorithm is possible. For more information, we refer to [4].

4 The solution of the $\underline{\text{P}}$VRIP

The algorithm **VRIP** solved the $\underline{\text{L}}$VRIP, but from the $m+1$ polynomial $(m+1)$-tuples $v_p^j(z)$, $j = 1, \ldots, m+1$, it constructs, we can now easily find a solution of the proper interpolation problem $\underline{\text{P}}$VRIP as described in the following theorem.

Theorem 3 *Let $v_p^j(z), j = 1, 2, \ldots, m+1$ be defined by theorem 2 and ordered with respect to increasing s-degree. Set as before $G_p(z) = [v_p^1(z)\cdots v_p^{m+1}(z)]$ the matrix containing these $(m+1)$-tuples. If $v_p^1(z)$ is not a solution of the $\underline{\text{P}}$VRIP, then $\sum_{k=1}^{l} c_k v_p^k(z)$ is, where l is the smallest value such that $[v_p^1(z)\cdots v_p^l(z)]$ (i.e. the first l columns of $G_p(z)$) evaluated for $z = z_i, i = 1, 2, \ldots, p$, have for each value i rank 1. The l-tuple $(c_1, c_2, \ldots, c_l)$ is not allowed to be an element of a specific $(l-1)$ dimensional subspace of $\mathbb{K}^l$ for each interpolation point z_i. The latter condition should prevent the proposed solution to be divisible by $z - z_i$.*

Proof. When $v_p^1(z)$ is divisible by $z - z_i$ for a certain $i, 1 \leq i \leq p$, then $v_p^1(z)$ is a solution of the $\underline{\text{L}}$VRIP but not of the $\underline{\text{P}}$VRIP. It is clear that $V_i(z)$ as defined in (12) has full rank $m+1$ if $z \neq z_i$ and has rank 1 if $z = z_i$. Indeed, z appears only in the factor $\prod_{k\neq j_i} E_k(z - z_i)$. Hence, $G_p(z) = V_0(z)V_1(z)\cdots V_p(z)$ has full rank $m+1$ if $z \neq z_i$ and has rank 1 if $z = z_i$. Therefore, to get a solution of smallest s-degree, we can take the linear polynomial combination $\sum_{k=1}^{l} c_k v_p^k(z)$ as a candidate where l is chosen as indicated above. For a smaller value of l, there is at least one z_i for which the first l columns of $G_p(z_i)$ would form a matrix of rank 0. Hence, we could not find a linear polynomial combination of the $v_p^k(z)$, $k = 1, 2, \ldots, l$, not divisible by $z - z_i$. However, with the value of l as chosen above,

$\sum_{k=1}^{l} c_k v_p^k(z)$ is not divisible by $z - z_i$, $i = 1, 2, \ldots$ as long as the l-tuple $(c_1, c_2, \ldots, c_l)$ isn't a solution of $G_p^l(z_i)[c_1 \ \ldots \ c_l]^T = 0$ with $G_p^l(z)$ the fist l columns of $G_p(z)$. We have chosen l such that $G_p^l(z_i)$ has rank 1. Hence, the null space has dimension $(l-1)$ and is the specific subspace of $\mathbb{K}^l$, we are referring to above. □

5 A parameterization of all solutions of the LVRIP and PVRIP

To describe all possible solutions of the linearized and proper interpolation problems, we can formulate the following theorems.

Theorem 4 *If the s-degree α_p^1 of $v_p^1(z)$ is smaller than the s-degree α_p^2 of $v_p^2(z)$, then there is only one rational function $n_p^1(z)/d_p^1(z)$ which solves the LVRIP. Otherwise a parameterization of all solutions is given by:*

$$\begin{bmatrix} n_p^c(z) \\ d_p^c(z) \end{bmatrix} = \begin{bmatrix} n_p^1 & n_p^2 & \cdots & n_p^l \\ d_p^1 & d_p^2 & \cdots & d_p^l \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_l \end{bmatrix}$$

with $c = (c_1, c_2, \ldots, c_l) \neq (0, 0, \ldots, 0) \in \mathbb{K}^l$ and l is such that $\alpha_p^1 = \alpha_p^2 = \cdots = \alpha_p^l < \alpha_p^{l+1}$ ($\alpha_p^{m+2} = +\infty$). If $c \neq kc'$ with $k \in \mathbb{K}$, then $n_p^c(z)/d_p^c(z)$ and $n_p^{c'}(z)/d_p^{c'}(z)$ are different vector rational functions.

Proof. That the rational approximant constructed is a solution of the LVRIP is trivial. We shall only prove the assertion of the last sentence. For the parameter l-tuple $c \neq kc'$, we set

$$\begin{bmatrix} n_p^c(z) & n_p^{c'}(z) \\ d_p^c(z) & d_p^{c'}(z) \end{bmatrix} = \begin{bmatrix} n_p^1(z) & n_p^2(z) & \cdots & n_p^l(z) \\ d_p^1(z) & d_p^2(z) & \cdots & d_p^l(z) \end{bmatrix} \begin{bmatrix} c_1 & c_1' \\ c_2 & c_2' \\ \vdots & \vdots \\ c_l & c_l' \end{bmatrix}.$$

Because the right-hand side of the previous equation has rank 2 for $z \neq z_i, i = 1, 2, \ldots, p$, we derive that the left-hand side has rank 2 for an infinite number of values for z. Hence, the two vector rational functions $n_p^c(z)/d_p^c(z)$ and $n_p^{c'}(z)/d_p^{c'}(z)$ are different. □

Concerning a parameterization of all solutions of the PVRIP, we have the following.

Theorem 5 *If $v_p^1(z)$ which solves the LVRIP is also a solution of the PVRIP, the two cases of the previous theorem also apply here, with the only exception that $(c_1, c_2, \ldots, c_l)$ is not allowed to be an element of a specific $(l-1)$-dimensional subspace of $\mathbb{K}^l$ connected to each interpolation point $z_i, i = 1, 2, \ldots, p$ (see theorem 3). If $v_p^1(z)$ is not a solution of the PVRIP, a parameterization of all solutions of the problem is given by:*

$$\begin{bmatrix} n_p^c(z) \\ d_p^c(z) \end{bmatrix} = \begin{bmatrix} n_p^1 & n_p^2 & \cdots & n_p^{l'} \\ d_p^1 & d_p^2 & \cdots & d_p^{l'} \end{bmatrix} \begin{bmatrix} c_1(z) \\ c_2(z) \\ \vdots \\ c_{l'}(z) \end{bmatrix}$$

with l' related to the l of theorem 3 by $\alpha_p^l = \alpha_p^{l+1} = \cdots = \alpha_p^{l'} < \alpha_p^{l'+1}$ and $\deg c_j(z) \le \alpha_p^l - \alpha_p^j, j = 1, 2, \ldots, l'$. Moreover $c = (c_1(z_i), c_2(z_i), \ldots, c_{l'}(z_i))$ is not allowed to be an element of a specific $(l'-1)$-dimensional subspace of $\mathbb{K}^{l'}$ for each interpolation point $z_i, i = 1, 2, \ldots, p$. As before we can show that, except for a scaling factor, for different polynomial l'-tuples $(c_1(z), c_2(z), \ldots, c_{l'}(z))$, we get different vector rational functions.

The proof goes along the same lines as the previous one, making additional use of the result of the previous section. We leave the details to the reader.

6 The confluent case

When for certain interpolation points z_i, not only the function value f_i is given but also the values of one or several consecutive derivatives, we get the confluent $\underline{\text{L}}$VRIP and $\underline{\text{P}}$VRIP. If we consider the interpolation data for the "confluent" points in the natural ordering, i.e., function value, first derivative value, second derivative value,..., then we can prove in a similar way that the same algorithm can be used to compute the polynomial $(m+1)$-tuples $v_p^j(z)$ of the confluent problem. The parameterization of all solutions of the $\underline{\text{L}}$VRIP and $\underline{\text{P}}$VRIP is also valid in this case. The completely confluent problem was discussed as a minimal vector Padé problem in [6], where we also developed a more general "nice" problem setting, allowing the original data to be known only partially leading to partially determined residuals. It is clear that this generalization can also be made for the vector rational interpolation problem investigated here, leading to similar algorithms. As a special case, we indicated in [6] that taking the shift parameters equal to each other leads to the vector Padé approximants defined by van Iseghem [21]. The general confluent problem where we consider possible interpolation data around the interpolation point ∞ and where an interpolation point can be a pole of the rational function, requires a much more careful setting of the problem. This shall be worked out for the vector problem in a future publication on a more formal basis, based on [20] handling the scalar case.

7 Example 1 revisited

We take the following interpolation data: $z_1 = 0, f_1 = [1\ 0]^T, z_2 = 1, f_2 = [2\ 1]^T, z_3 = 2, f_3 = [2\ 1]^T$. We use the Chebyshev polynomials $T_k(z)$ as a basis for the set of polynomials: $T_0(z) = 1, T_1(z) = z$ and $T_{k+1}(z) = 2zT_k(z) - T_{k-1}(z)$ for $k \ge 1$. Hence, the multiplication of $T_k(z)$ by $z - z_{p+1}$ can be written as $(z - z_{p+1})T_0(z) = T_1(z) - z_{p+1}T_0(z)$ and $(z - z_{p+1})T_k(z) = T_{k+1}(z)/2 - z_{p+1}T_k(z) + T_{k-1}(z)/2$ for $k \ge 1$. The algorithm of section 3 for $s = (s_1, s_2) = (0, 1)$ generates the following $V_i(z)$-matrices:

$$V_0 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, V_1 = \begin{bmatrix} z & 0 & 0 \\ 0 & 1 & z \\ 0 & 1 & 0 \end{bmatrix}, V_2 = \begin{bmatrix} z-1 & 0 & 1 \\ 0 & z-1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, V_3 = \begin{bmatrix} 0 & z-2 & 0 \\ 1 & 0 & z-2 \\ 1 & 0 & 0 \end{bmatrix}.$$

We derive

$$\begin{aligned} G_3(z) &= V_0(z)V_1(z)V_2(z)V_3(z) \\ &= \begin{bmatrix} v_3^1(z) & v_3^2(z) & v_3^3(z) \end{bmatrix} \\ &= \begin{bmatrix} 2T_1(z) & 0 & \frac{1}{2}T_2(z) - 3T_1(z) + \frac{5}{2}T_0(z) \\ T_1(z) & \frac{1}{4}T_3(z) - \frac{3}{2}T_2(z) + \frac{11}{4}T_1(z) - \frac{3}{2}T_0(z) & 0 \\ T_1(z) & 0 & \frac{1}{2}T_2(z) - 3T_1(z) + \frac{5}{2}T_0(z) \end{bmatrix}. \end{aligned}$$

Therefore, the vector rational function solving the LVRIP is unique and equal to

$$n_3^1(z)/d_3^1(z) = \begin{bmatrix} 2T_1(z) \\ T_1(z) \\ T_1(z) \end{bmatrix}.$$

This solution has minimal s-degree 1. However, it is not a solution of the PVRIP, because $T_1(z) = z$ is a common factor of the numerator and denominator polynomials. Hence, 0 is an "unattainable point". Theorem 5 gives all solutions of the PVRIP having a minimal s-degree 2, as

$$v_3^a(z) = \begin{bmatrix} v_3^1(z) & v_3^2(z) & v_3^3(z) \end{bmatrix} \begin{bmatrix} a_1(z) \\ a_2 \\ a_3 \end{bmatrix}$$

with $\deg a_1(z) \leq 1$ and $G_3(z_i)[a_1(z_i)\ a_2\ a_3]^T \neq 0$ (making $v_3^a(z)$ not divisible $z - z_i$) or $\deg a_1(z) \leq 1$, $a_1(1) \neq 0$, $a_1(2) \neq 0$ and $a_3 \neq 0$.

8 Conclusion

In this short paper we have described how the ideas of rational interpolation without reordering the interpolation points can be generalized to the case where vector data are given. The approximant is minimal with respect to the s-degree complexity measure. An efficient algorithm, allowing several possible implementations is given, yet it describes precisely what operations have to be performed. The algorithm finds a solution which solves a linearized interpolation problem. However, it is shown how the results of the algorithm can be used to generate not only all the possible solutions of this problem but also how all the solutions for the proper rational interpolation problem can be constructed.

The treatment is kept as simple as reasonably possible to make the ideas clear without being troubled by exceptional cases and too much notational burden. For this reason we supposed in the main development that all the interpolation points are different. However, as we mentioned in section 6, the results for the confluent case are basically the same. In the case where all the interpolation points coincide, we recover the minimal vector Padé approximants of [6]. For the same reason of simplicity, we did not allow the interpolation points to be at infinity or prescribed poles at the interpolation points. For the scalar problem, all these generalizations and more elaborate proofs are given in [20]. The vector case will be solved along the same lines in a future publication.

References

[1] A.C. Antoulas. Rational interpolation and the Euclidean algorithm. *Lin. Alg. Appl.*, 108:157–171, 1988.

[2] A.C. Antoulas and B.D.O. Anderson. On the scalar rational interpolation problem. *IMA J. Math. Control and Information*, 3:61–88, 1986.

[3] A.C. Antoulas and B.D.O. Anderson. On the problem of stable rational interpolation. *Lin. Alg. Appl.*, 122/123/124:301–329, 1989.

[4] A. Bruckstein and T. Kailath. An inverse scattering framework for several problems in signal processing. *IEEE Acoust. Speech Signal Process. magazine*, pages 6–20, 1987.

[5] A. Bultheel. *Laurent series and their Padé Approximations*, volume OT27 of *Operator theory: Advances and Applications*. Birkhäuser-Verlag, Basel, 1987.

[6] A. Bultheel and M. Van Barel. Minimal vector Padé approximation. *J. Comp. Appl. Math.*, September 1990.

[7] I. Gohberg, T. Kailath, I. Koltracht, and P. Lancaster. Linear complexity parallel algorithms for linear systems of equations with recursive structure. *Lin. Alg. Appl.*, 88:271–315, 1987.

[8] P.R. Graves-Morris. Practical, reliable, rational interpolation. *J. Inst. Maths. Applics.*, 25:267–286, 1980.

[9] P.R. Graves-Morris. Efficient reliable rational interpolation. In H. van Rossum M. de Bruin, editor, *Padé Approximation and its Applications*, volume 888 of *Lecture Notes in Math.*, Berlin, 1981. Springer Verlag.

[10] P.R. Graves-Morris. Vector valued interpolants I. *Numerische Mathematik*, 42:331–348, 1983.

[11] P.R. Graves-Morris. Vector valued interpolants II. *IMA J. Numer. Anal.*, 4:209–224, 1984.

[12] P.R. Graves-Morris and T.R. Hopkins. Reliable rational interpolation. *Numer. Math.*, 36:111–128, 1981.

[13] P.R. Graves-Morris and C.D. Jenkins. Vector valued interpolants III. *Constructive Approx.*, 2:263–289, 1986.

[14] M.H. Gutknecht. Continued fractions associated with the Newton-Padé table. *Numer. Math.*, 56:547–589, 1989.

[15] M.H. Gutknecht. The rational interpolation problem revisited. *Rocky Mountain Journal of Mathematics*, 21(1):263–280, 1991.

[16] G. Heinig, W. Hoppe, and K. Rost. Structured matrices in interpolation and approximation problems. *Wiss. Z. d. TU Karl-Marx-Stadt*, 31:196–202, 1989.

[17] M. Van Barel. *Nested Minimal Partial Realizations and Related Matrix Rational Approximants*. PhD thesis, K.U. Leuven, January 1989.

[18] M. Van Barel and A. Bultheel. Minimal Padé sense matrix approximations around $s = 0$ and $s = \infty$. In A. Cuyt, editor, *Nonlinear numerical methods and rational approximation*, pages 143–154, Dordrecht, 1989. Reidel Publ. Comp.

[19] M. Van Barel and A. Bultheel. A new approach to the rational interpolation problem. *J. Comp. Appl. Math.*, 32, 1990. Accepted.

[20] M. Van Barel and A. Bultheel. A new formal approach to the rational interpolation problem. Technical Report TW134, Department of Computer Science, K.U.Leuven, July 1990. (35 pages).

[21] J. van Iseghem. *Approximants de Padé vectoriels*. PhD thesis, Lille, 1987.

[22] H. Werner. A reliable method for rational interpolation. In L. Wuytack, editor, *Padé approximation and its applications*, pages 257–277, Berlin-Heidelberg-New York, 1979. Springer Verlag.

[23] H. Werner. Ein Algorithmus zur rationalen Interpolation. In H. Werner L. Collatz, G. Meinardus, editor, *Numerische Methoden der Approximationstheorie, 5*, pages 319–337, Basel, 1980. Birkhäuser-Verlag. ISNM 52.

[24] H. Werner. Altes und Neues zur rationalen Interpolation, September 1981. Vortrag 10 Österreichischen Mathematiker Kongress, Insbruck.

[25] H. Werner. Algorithm 51 : A reliable and numerically stable program for rational interpolation of Lagrange data. *Computing*, 31:269–286, 1983.