

Multivariate Frobenius-Padé approximants: properties and algorithms

Ana C. Matos

Laboratoire Paul Painlevé UMR CNRS 8524
UFR de Mathématiques Pures et Appliquées
Université des Sciences et Technologies de Lille, France
e-mail: matos@math.univ-lille1.fr *

July 2006

Abstract

The aim of this paper is to construct rational approximants for multivariate functions given by their expansion in an orthogonal polynomial system. This will be done by generalizing the concept of multivariate Padé approximation. After defining the *multivariate Frobenius-Padé approximants*, we will be interested in the two following problems: the first one is to develop recursive algorithms for the computation of the value of a sequence of approximants at a given point. The second one is to compute the coefficients of the numerator and denominator of the approximants by solving a linear system. For some particular cases we will obtain a displacement rank structure for the matrix of the system we have to solve. The case of a Tchebyshev expansion is considered in more detail.

Keywords: multivariate Padé approximants, Tchebyshev series, orthogonal polynomials, orthogonal expansions, rational approximation, displacement rank structure

Subject Classification: AMS(MOS): 15A99, 33C45, 41A20, 41A21, 41A63, 42C05, 42C10, 65F05, 65F30.

1 Introduction

Let us consider a two variable function $f(x, y)$ given by its expansion (or the first coefficients of its expansion) in an orthogonal polynomial system $\{P_k\}$

$$f(x, y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} c_{ij} P_i(x) P_j(y).$$

*This work was partly supported by INTAS Research Network 03-51-6637

We want to construct rational approximants for f by generalizing the concept and ideas of Padé approximation - rational approximation for power series (see for instance [1] and [2])

The univariate case has been studied in [22]. There the *Padé-Legendre approximants* for a series $f(x) = \sum_{i=0}^{\infty} c_i P_i(x)$ have been defined and different types of algorithms for their recursive computation have been proposed. For some classes of functions, acceleration results have been obtained, that is, it has been shown that the sequence of approximants converges to f faster than the partial sums. The numerical examples were very good and incited us to generalize these ideas to the case of a vector function. The *simultaneous Frobenius-Padé approximants* were defined in [23] where their properties have been given and recurrence relations between adjacent approximants in a Frobenius-Padé table have been established, together with recursive algorithms for their computation.

We are now going to generalize the ideas of Frobenius-Padé approximation to the multivariate case (we will restrict ourselves to the two variable case). Different generalizations of Padé approximation to the multivariate case have been proposed and inspired our work. Different approaches have been developed by A. Cuyt [7, 8, 9, 10, 11], P.Guillaume [13, 14], and other authors (see for instance [19], [21], [6], ..). Properties of the different approximants, convergence properties and ways of computing them have been developed. We take advantage of these different ideas to generalize them to the case of orthogonal expansions.

After giving the general definition of the *multivariate Frobenius-Padé approximants*, we will be interested in the way how to compute them. We will consider two different situations:

- compute the values of a sequence of approximants at a given point (x_0, y_0) ;
- compute the coefficients of the denominators and numerators of a sequence of approximants.

We will see that these computations are equivalent to the solution of linear systems.

In order to define an approximant we will need to choose three sets of indices: the set of indices appearing in the numerator which we will denote by N , the set of indices appearing in the denominator D and the one corresponding to the terms that will be annihilated in the error term, E . We have also many ways of defining sequences of approximants: fixing the denominator (numerator) and increasing the cardinal of the set of indices in N (respectively D), increasing simultaneously the sets N and D ... Our aim in this paper is to obtain, for particular choices in this large variety of parameters, either recursive algorithms to compute a sequence of approximations, either a displacement rank structure for the matrix of the system we have to solve to obtain the approximant.

A second generalization of Frobenius-Padé approximation based on the ideas of [13] for Padé approximants is then developed- the *mixed Frobenius-Padé approximants*. We will consider the particular case of a Tchebyshev series for which the computation of the coefficient matrix of the system which gives the denominator coefficients of the approximant is very simple and we will show that the matrix has in this case a block Toeplitz-plus-Hankel structure.

Let us begin with the first definitions.

2 Definition of the approximants

Let us consider a two variable function given by its expansion in an orthogonal series

$$f(x, y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} c_{ij} P_i(x) P_j(y) \quad (1)$$

where

$$\left\{ \begin{array}{l} \{P_i\} \text{ is a system of orthogonal polynomials in } [a, b] \text{ with respect to} \\ \text{the weight function } w \\ \gamma_{ij} = \|P_i\| \|P_j\|, \quad \|P\|^2 = \int_a^b P(x)^2 w(x) dx \\ c_{ij} = \frac{1}{\gamma_{ij}} \int_a^b \int_a^b f(x, y) P_i(x) P_j(y) w(x) w(y) dx dy \end{array} \right.$$

We search for two polynomials $P(x, y)$ and $Q(x, y)$

$$\left\{ \begin{array}{l} P(x, y) = \sum_{(i,j) \in N} a_{ij} P_i(x) P_j(y) \\ Q(x, y) = \sum_{(i,j) \in D} b_{ij} P_i(x) P_j(y) \end{array} \right. \quad (2)$$

satisfying

$$f(x, y)Q(x, y) - P(x, y) = \sum_{(i,j) \in (\mathbb{N}^2 \setminus E)} d_{ij} P_i(x) P_j(y) \quad (3)$$

where

- $D \subset \mathbb{N}^2$ has m elements $(k_1, l_1), \dots, (k_m, l_m)$;
- E is the set of indices of the null terms in the expansion of the remainder, $N \subset E$ with n elements $(i_1, j_1), (i_2, j_2), \dots, (i_n, j_n)$, $\text{card}(N) = n$;
- $H = E \setminus N$ with $m - 1$ elements $(i_{n+1}, j_{n+1}), \dots, (i_{n+m-1}, j_{n+m-1})$.

We then define the **multivariate Frobenius-Padé approximant of the function f** as the rational function

$$R(x, y) = \frac{P(x, y)}{Q(x, y)}$$

We set, for $(k, l) \in \mathbb{N}^2$

$$P_k(x) P_l(y) f(x, y) = \sum_{(i,j) \in \mathbb{N}^2} h_{ij}^{kl} P_i(x) P_j(y) \quad (4)$$

and so we get

$$\begin{aligned} (fQ - P)(x, y) &= \sum_{(i,j) \in D} b_{ij} P_i(x) P_j(y) f(x, y) - \sum_{(i,j) \in N} a_{ij} P_i(x) P_j(y) \\ &= \sum_{(i,j) \in \mathbb{N}^2} \left(\sum_{(k,l) \in D} b_{kl} h_{ij}^{kl} \right) P_i(x) P_j(y) - \sum_{(i,j) \in N} a_{ij} P_i(x) P_j(y). \end{aligned}$$

Condition (3) writes

$$\sum_{(k,l) \in D} b_{kl} h_{ij}^{kl} = 0, \quad (i, j) \in E \setminus N \quad (5)$$

$$\sum_{(k,l) \in D} b_{kl} h_{ij}^{kl} = a_{ij}, \quad (i, j) \in N \quad (6)$$

As $\text{card}(D) = \text{card}(H) + 1$, system (5) is an homogeneous system of $(m - 1)$ equations and m unknowns, so it always has a nontrivial solution $(b_{kl})_{(k,l) \in D}$ (the coefficients of the denominator polynomial). Then the coefficients $(a_{ij})_{(i,j) \in N}$ of the numerator polynomial are immediately given by (6). So the most important part of the computational effort to compute a multivariate Frobenius-Padé approximant is the solution of system (5). The coefficients of the system matrix are the quantities (h_{ij}^{kl}) defined by (4). They can be computed recursively using the three term recurrence relation for the orthogonal system $\{P_k\}_{k \geq 0}$ as will be shown in the next paragraph.

2.1 Recursive computation of the (h_{ij}^{kl}) .

Let us consider the three term recurrence relation for the system of orthogonal polynomials $\{P_k\}_{k \geq 0}$

$$\begin{aligned} zP_i(z) &= \alpha_i P_{i+1}(z) + \beta_i P_i(z) + \gamma_i P_{i-1}(z) \quad i \geq 0 \quad (\text{we set } \gamma_{-1} = 0) \\ P_0(z) &= 1, \quad P_1(z) = (z - \beta_0)/\alpha_0 \end{aligned}$$

Then we have

$$P_{k+1}(x)P_l(y)f(x, y) = \frac{1}{\alpha_k} x P_k(x)P_l(y)f(x, y) - \frac{\beta_k}{\alpha_k} P_k(x)P_l(y)f(x, y) - \frac{\gamma_k}{\alpha_k} P_{k-1}(x)P_l(y)f(x, y) \quad (7)$$

But

$$\begin{aligned} xP_k(x)P_l(y)f(x, y) &= \\ &= \sum_{(i,j) \in \mathbb{N}^2} h_{ij}^{kl} (\alpha_i P_{i+1}(x) + \beta_i P_i(x) + \gamma_i P_{i-1}(x)) P_j(y) = \\ &= \sum_{(i,j) \in \mathbb{N}^2} (\alpha_{i-1} h_{i-1,j}^{kl} + \beta_i h_{ij}^{kl} + \gamma_{i+1} h_{i+1,l}^{kl}) P_i(x) P_j(y) \end{aligned}$$

Equating coefficients of the same type in equation (7) we get

$$\alpha_k h_{ij}^{k+1,l} = \alpha_{i-1} h_{i-1,j}^{kl} + \beta_i h_{ij}^{kl} + \gamma_{i+1} h_{i+1,l}^{kl} - \beta_k h_{ij}^{kl} - \gamma_k h_{ij}^{k-1,l} \quad (8)$$

and by symmetry with respect to the y variable

$$\alpha_l h_{ij}^{k,l+1} = \alpha_{j-1} h_{i,j-1}^{kl} + \beta_j h_{ij}^{kl} + \gamma_{j+1} h_{i,j+1}^{kl} - \beta_l h_{ij}^{kl} - \gamma_l h_{ij}^{k,l-1} \quad (9)$$

From these two equations, and giving suitable initializations, we can compute recursively all the quantities h_{ij}^{kl} from the initial data $h_{ij}^{00} = c_{ij}$, $i, j \geq 0$ (recursion is done on the superscripts $k, l \geq 0$). We also remark that, as by definition we have

$$h_{ij}^{kl} = \frac{1}{\gamma_{ij}} \int_a^b \int_a^b P_k(x)P_l(y)f(x, y)P_i(x)P_j(y)w(x)w(y)dx dy \quad \text{with } \gamma_{ij} = \|P_i\|^2 \|P_j\|^2,$$

then

$$h_{ij}^{kl} \gamma_{ij} = h_{kl}^{ij} \gamma_{kl}.$$

This implies that we only need to compute h_{ij}^{kl} for $k \leq i, l \leq j (i, j \geq 0)$.

We can then construct the matrix

$$\mathcal{H} = \begin{pmatrix} h_{i_{n+1}j_{n+1}}^{k_1l_1} & h_{i_{n+1}j_{n+1}}^{k_2l_2} & \cdots & h_{i_{n+1}j_{n+1}}^{k_m l_m} \\ \vdots & \vdots & \cdots & \vdots \\ h_{i_{n+m-1}j_{n+m-1}}^{k_1l_1} & h_{i_{n+m-1}j_{n+m-1}}^{k_2l_2} & \cdots & h_{i_{n+m-1}j_{n+m-1}}^{k_m l_m} \end{pmatrix} \quad (10)$$

So the problem of computing the denominator coefficients of one approximant is equivalent to the solution of the linear system (5). If $\text{rank}(\mathcal{H}) = m - 1$ then the solution is unique apart from a multiplicative factor (we need to fix the normalization of the denominator polynomial).

But we can consider different kinds of approximation problems: instead of computing one Frobenius-Padé approximant, one may be interested in computing the value in a given point of a sequence of approximants. For this we will develop in the next section a recursive algorithm.

3 Recursive algorithm

Using Cramer's rule for solving the systems (6) and (5) we immediately get the following determinantal expression for the numerator and the denominator of the Frobenius-Padé approximant:

$$Q(x, y) = \begin{vmatrix} P_{k_1}(x)P_{l_1}(y) & P_{k_2}(x)P_{l_2}(y) & \cdots & P_{k_m}(x)P_{l_m}(y) \\ h_{i_{n+1}j_{n+1}}^{k_1l_1} & h_{i_{n+1}j_{n+1}}^{k_2l_2} & \cdots & h_{i_{n+1}j_{n+1}}^{k_m l_m} \\ h_{i_{n+2}j_{n+2}}^{k_1l_1} & h_{i_{n+2}j_{n+2}}^{k_2l_2} & \cdots & h_{i_{n+2}j_{n+2}}^{k_m l_m} \\ \vdots & \vdots & \vdots & \vdots \\ h_{i_{n+m-1}j_{n+m-1}}^{k_1l_1} & h_{i_{n+m-1}j_{n+m-1}}^{k_2l_2} & \cdots & h_{i_{n+m-1}j_{n+m-1}}^{k_m l_m} \end{vmatrix}$$

$$P(x, y) = \sum_{(i,j) \in N} a_{ij} P_i(x) P_j(y) = \sum_{(i,j) \in N} \left(\sum_{(k,l) \in D} b_{kl} h_{ij}^{kl} \right) P_i(x) P_j(y) =$$

$$= \begin{vmatrix} \sum_{(i,j) \in N} h_{ij}^{k_1l_1} P_i(x) P_j(y) & \sum_{(i,j) \in N} h_{ij}^{k_2l_2} P_i(x) P_j(y) & \cdots & \sum_{(i,j) \in N} h_{ij}^{k_m l_m} P_i(x) P_j(y) \\ h_{i_{n+1}j_{n+1}}^{k_1l_1} & h_{i_{n+1}j_{n+1}}^{k_2l_2} & \cdots & h_{i_{n+1}j_{n+1}}^{k_m l_m} \\ h_{i_{n+2}j_{n+2}}^{k_1l_1} & h_{i_{n+2}j_{n+2}}^{k_2l_2} & \cdots & h_{i_{n+2}j_{n+2}}^{k_m l_m} \\ \vdots & \vdots & \vdots & \vdots \\ h_{i_{n+m-1}j_{n+m-1}}^{k_1l_1} & h_{i_{n+m-1}j_{n+m-1}}^{k_2l_2} & \cdots & h_{i_{n+m-1}j_{n+m-1}}^{k_m l_m} \end{vmatrix}.$$

Following the ideas developed in [7] for the recursive computation of multivariate Padé approximants, we will do some row and column manipulation in these two determinants. Let us divide the first column by $P_{k_1}(x)P_{l_1}(y)$, \cdots , the m th column by $P_{k_m}(x)P_{l_m}(y)$; then we multiply the

second row by $P_{i_{n+1}}(x)P_{j_{n+1}}(y), \dots$, the m th row by $P_{i_{n+m-1}}(x)P_{j_{n+m-1}}(y)$. We obtain:

$$\frac{P(x, y)}{Q(x, y)} = \frac{\begin{vmatrix} \frac{\sum_{(i,j) \in N} h_{ij}^{k_1 l_1} P_i(x) P_j(y)}{P_{k_1}(x) P_{l_1}(y)} & \cdots & \frac{\sum_{(i,j) \in N} h_{ij}^{k_m l_m} P_i(x) P_j(y)}{P_{k_m}(x) P_{l_m}(y)} \\ h_{i_{n+1} j_{n+1}}^{k_1 l_1} \frac{P_{i_{n+1}}(x) P_{j_{n+1}}(y)}{P_{k_1}(x) P_{l_1}(y)} & \cdots & h_{i_{n+1} j_{n+1}}^{k_m l_m} \frac{P_{i_{n+1}}(x) P_{j_{n+1}}(y)}{P_{k_m}(x) P_{l_m}(y)} \\ \vdots & \vdots & \vdots \\ h_{i_{n+m-1} j_{n+m-1}}^{k_1 l_1} \frac{P_{i_{n+m-1}}(x) P_{j_{n+m-1}}(y)}{P_{k_1}(x) P_{l_1}(y)} & \cdots & h_{i_{n+m-1} j_{n+m-1}}^{k_m l_m} \frac{P_{i_{n+m-1}}(x) P_{j_{n+m-1}}(y)}{P_{k_m}(x) P_{l_m}(y)} \end{vmatrix}}{\begin{vmatrix} 1 & \cdots & 1 \\ h_{i_{n+1} j_{n+1}}^{k_1 l_1} \frac{P_{i_{n+1}}(x) P_{j_{n+1}}(y)}{P_{k_1}(x) P_{l_1}(y)} & \cdots & h_{i_{n+1} j_{n+1}}^{k_m l_m} \frac{P_{i_{n+1}}(x) P_{j_{n+1}}(y)}{P_{k_m}(x) P_{l_m}(y)} \\ \vdots & \vdots & \vdots \\ h_{i_{n+m-1} j_{n+m-1}}^{k_1 l_1} \frac{P_{i_{n+m-1}}(x) P_{j_{n+m-1}}(y)}{P_{k_1}(x) P_{l_1}(y)} & \cdots & h_{i_{n+m-1} j_{n+m-1}}^{k_m l_m} \frac{P_{i_{n+m-1}}(x) P_{j_{n+m-1}}(y)}{P_{k_m}(x) P_{l_m}(y)} \end{vmatrix}}$$

Let us set

$$\begin{cases} s_u(n) = \sum_{r=1}^n h_{i_r j_r}^{k_u l_u} P_{i_r}(x) P_{j_r}(y) / P_{k_u}(x) P_{l_u}(y) \\ \Delta s_u(n) = s_u(n+1) - s_u(n) \end{cases}, \quad u = 1, \dots, m, n \geq 0.$$

Then

$$\frac{P(x, y)}{Q(x, y)} = \frac{\begin{vmatrix} s_1(n) & \cdots & s_m(n) \\ \Delta s_1(n) & \cdots & \Delta s_m(n) \\ \vdots & \vdots & \vdots \\ \Delta s_1(n+m-2) & \cdots & \Delta s_m(n+m-2) \end{vmatrix}}{\begin{vmatrix} 1 & \cdots & 1 \\ \Delta s_1(n) & \cdots & \Delta s_m(n) \\ \vdots & \vdots & \vdots \\ \Delta s_1(n+m-2) & \cdots & \Delta s_m(n+m-2) \end{vmatrix}}$$

Again with some column and row manipulation we get

$$\begin{aligned}
& \frac{P(x, y)}{Q(x, y)} = \\
& = \frac{\begin{vmatrix} s_1(n) & \cdots & s_m(n) \\ s_1(n+1) & \cdots & s_m(n+1) \\ \vdots & \vdots & \vdots \\ s_1(n+m-1) & \cdots & s_m(n+m-1) \end{vmatrix}}{\begin{vmatrix} 1 & 0 & \cdots & 0 \\ \Delta s_1(n) & \Delta s_2(n) - \Delta s_1(n) & \cdots & \Delta s_m(n) - \Delta s_{m-1}(n) \\ \vdots & \vdots & \vdots & \vdots \\ \Delta s_1(n+m-2) & \Delta s_2(n+m-2) - \Delta s_1(n+m-2) & \cdots & \Delta s_m(n+m-2) - \Delta s_{m-1}(n+m-1) \end{vmatrix}} \\
& = \frac{\begin{vmatrix} s_1(n) & s_1(n+1) & \cdots & s_1(n+m-1) \\ s_2(n) & s_2(n+1) & \cdots & s_2(n+m-1) \\ \vdots & \vdots & \vdots & \vdots \\ s_m(n) & s_m(n+1) & \cdots & s_m(n+m-1) \end{vmatrix}}{\begin{vmatrix} \Delta s_2(n) - \Delta s_1(n) & \cdots & \Delta s_m(n) - \Delta s_{m-1}(n) \\ \vdots & \vdots & \vdots \\ \Delta s_2(n+m-2) - \Delta s_1(n+m-2) & \cdots & \Delta s_m(n+m-2) - \Delta s_{m-1}(n+m-2) \end{vmatrix}}
\end{aligned}$$

Finally we set

$$g_i(n) = s_{i+1}(n) - s_i(n), \quad i = 1, 2, \dots, m-1$$

and then

$$P(x, y) = \begin{vmatrix} s_1(n) & s_1(n+1) & \cdots & s_1(n+m-1) \\ g_1(n) & g_1(n+1) & \cdots & g_1(n+m-1) \\ \vdots & \vdots & \vdots & \vdots \\ g_{m-1}(n) & g_{m-1}(n+1) & \cdots & g_{m-1}(n+m-1) \end{vmatrix}$$

$$\begin{aligned}
Q(x, y) &= \\
&= \begin{vmatrix} g_1(n+1) - g_1(n) & \cdots & g_{m-1}(n+1) - g_{m-1}(n) \\ \vdots & \vdots & \vdots \\ g_1(n+m-1) - g_1(n+m-2) & \cdots & g_{m-1}(n+m-1) - g_{m-1}(n+m-2) \end{vmatrix} \\
&= \begin{vmatrix} 1 & g_1(n) & \cdots & g_{m-1}(n) \\ 0 & \Delta g_1(n) & \cdots & \Delta g_{m-1}(n) \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \Delta g_1(n+m-2) & \cdots & \Delta g_{m-1}(n+m-2) \end{vmatrix} = \\
&= \begin{vmatrix} 1 & g_1(n) & \cdots & g_{m-1}(n) \\ 1 & g_1(n+1) & \cdots & g_{m-1}(n+1) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & g_1(n+m-1) & \cdots & g_{m-1}(n+m-1) \end{vmatrix}
\end{aligned}$$

Finally

$$\frac{P(x, y)}{Q(x, y)} = \frac{\begin{vmatrix} s_1(n) & s_1(n+1) & \cdots & s_1(n+m-1) \\ g_1(n) & g_1(n+1) & \cdots & g_1(n+m-1) \\ \vdots & \vdots & \vdots & \vdots \\ g_{m-1}(n) & g_{m-1}(n+1) & \cdots & g_{m-1}(n+m-1) \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ g_1(n) & g_1(n+1) & \cdots & g_1(n+m-1) \\ \vdots & \vdots & \vdots & \vdots \\ g_{m-1}(n) & g_{m-1}(n+1) & \cdots & g_{m-1}(n+m-1) \end{vmatrix}} \quad (11)$$

It is well-known that this quotient of determinants can be computed by the E-algorithm [3] with the initializations:

$$\begin{cases} E_0^{(n)} = s_1(n) & \mathbb{N} \in \mathbb{N} \\ g_{0,i}^{(n)} = g_i(n) & i = 1, 2, \dots \end{cases}$$

and the recursive rules:

$$\begin{cases} E_k^{(n)} = E_{k-1}^{(n)} - \frac{E_{k-1}^{(n+1)} - E_{k-1}^{(n)}}{g_{k-1,k}^{(n+1)} - g_{k-1,k}^{(n)}} g_{k-1,k}^{(n)} \\ g_{k,i}^{(n)} = g_{k-1,i}^{(n)} - \frac{g_{k-1,i}^{(n+1)} - g_{k-1,i}^{(n)}}{g_{k-1,k}^{(n+1)} - g_{k-1,k}^{(n)}} g_{k-1,k}^{(n)}, & i = k+1, k+2, \dots \end{cases}$$

We conclude that the value at (x, y) of this Frobenius-Padé approximant $P(x, y)/Q(x, y)$ can be computed by the E-algorithm

This enables us to compute the value at a fixed point (x, y) of a sequence of approximants. For this let us consider two sequences of indices:

- $\{(i_1, j_1), (i_2, j_2), \dots, (i_n, j_n), \dots\}$ which corresponds to the indices of the polynomials $P_{i_u}(x)P_{j_u}(y)$ appearing in the numerators plus the indices of the coefficients that will be cancelled out in the remainder term;
- $\{(k_1, l_1), (k_2, l_2), \dots, (k_m, l_m), \dots\}$ which corresponds to the indices of the polynomials $P_{k_u}(x)P_{l_u}(y)$ appearing in the denominator.

We set for $n, m \geq 0$, $N_n = \{(i_1, j_1), (i_2, j_2), \dots, (i_n, j_n)\}$ and $D_m = \{(k_1, l_1), (k_2, l_2), \dots, (k_m, l_m)\}$ and define for $n \geq 0, m \geq 0$ the multivariate Frobenius-Padé approximant

$$S_{n,m}(x, y) = P(x, y)/Q(x, y)$$

by (2) and (3) with $N = N_n$, $D = D_m$ and $E = N_{n+m}$

Applying the E-algorithm with the initializations

$$\begin{cases} E_0^{(u)} &= \frac{\sum_{(i,j) \in N_u} h_{ij}^{k_1 l_1} P_i(x) P_j(y)}{P_{k_1}(x) P_{l_1}(y)} \quad u = 0, 1, \dots \\ g_{0,v}^{(u)} &= \frac{\sum_{(i,j) \in N_u} h_{ij}^{k_v+1 l_v+1} P_i(x) P_j(y)}{P_{k_{v+1}}(x) P_{l_{v+1}}(y)} - \frac{\sum_{(i,j) \in N_u} h_{ij}^{k_v l_v} P_i(x) P_j(y)}{P_{k_v}(x) P_{l_v}(y)} \quad u, v = 0, 1, \dots \end{cases}$$

we get

$$S_{n,m}(x, y) = E_m^{(n)} \text{ for } m, n \geq 0.$$

We can also apply the particular rules of the E-algorithm given in [[4]] which enable us to compute the quantities $S_{n,m+k}(x, y)$ directly from the quantities $S_{n,m}(x, y), \dots, S_{n+k,m}(x, y)$ (see [4] for details). These rules can be interesting from a numerical point of view to prevent instabilities.

If we dispose these approximants in a double entry table as usual with the Padé table, we have a recursive algorithm to compute the value in a given point of this double sequence of approximants. So we know how to solve the so called “value problem”.

But we can be interested in obtaining an explicit form for the approximant - the so-called “coefficient problem”. In this case we will need to solve, for each approximant a linear system. We will try, in the following sections, by defining particular sequences in the table or choosing a particular family of orthogonal polynomials (namely the Tchebyshev polynomials), to obtain some recursive algorithm to compute the coefficients of a sequence of approximants.

4 Different choices of sequences of approximants.

In the general definition of the multivariate Frobenius-Padé approximants (2) and (3), the sets of indices N, D, E indicating the indices of the polynomials present in the numerator, denominator and remainder term only need to satisfy the conditions:

$$\text{card}(E) = \text{card}(D) + \text{card}(N) - 1, \quad N \subset E, \quad (12)$$

and so they can be very general sets. In order to obtain recursive algorithms and convergence results we need to consider particular sequences of approximants, which means different choices for these index sets. We can define particular sequences in such a way that:

- a symmetry between the two variables x and y is preserved;
- the computation of the $m + 1$ term of the sequence can take profit of the computations done at the previous step.

We recall that from (5), if we order the indices in N, D, E in the following way:

$$\begin{aligned} N &= \{(i_1, j_1), \dots, (i_n, j_n)\} \\ D &= \{(k_1, l_1), \dots, (k_m, l_m)\} \\ E &= N \cup \{(u_1, v_1), \dots, (u_{m-1}, v_{m-1})\} \end{aligned} ,$$

the linear system we have to solve to obtain the denominator coefficients is

$$\begin{pmatrix} h_{u_1 v_1}^{k_1 l_1} & \dots & h_{u_1 v_1}^{k_m l_m} \\ \vdots & \dots & \vdots \\ h_{u_{m-1} v_{m-1}}^{k_1 l_1} & \dots & h_{u_{m-1} v_{m-1}}^{k_m l_m} \end{pmatrix} = \begin{pmatrix} b_{k_1 l_1} \\ \vdots \\ b_{k_m l_m} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \quad (13)$$

We will consider two main choices.

4.1 Tensor product type approximants

In this case, the sets N, D, E verifying (12) are of the following type:

- $E = \{(i, j) : 0 \leq i \leq m-1, 0 \leq j \leq m-1\}$, with $\text{card}(E) = m^2$.
- $N = \{(i, j) : 0 \leq i \leq m-l, 0 \leq j \leq m-l\}$
- $D = \{(i, j) : (m-l < i \leq m-1 \text{ and } 0 \leq j \leq m-1) \text{ or } (0 \leq i \leq m-1 \text{ and } m-l < j \leq m-1)\}$

for given m and l . We can also interchange the definitions of N and D . We get the approximants of the form

$$\mathcal{T}_{m,l}(x, y) = \frac{\sum_{i=0}^{m-l} \sum_{j=0}^{m-l} a_{ij} P_i(x) P_j(y)}{\sum_{i=0}^{m-1} \sum_{j=m-l}^{m-1} b_{ij} P_i(x) P_j(y) + \sum_{i=m-l}^{m-1} \sum_{j=0}^{m-1} b_{ij} P_i(x) P_j(y)}.$$

We can then consider the following sequences

1. **Vertical sequences:** let us consider

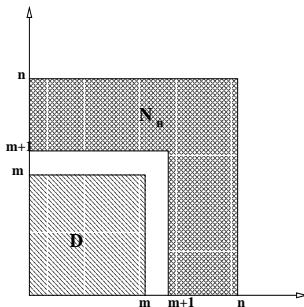
$$\begin{aligned} D &= \{(i, j) : 0 \leq i, j \leq m\} \text{ fixed} \\ N_n &= \{(i, j) : (m+1 \leq i \leq n, 0 \leq j \leq n) \text{ or } (0 \leq i \leq m, m+1 \leq j \leq n)\} \cup \{(0, 0)\} \\ E_n &= D \cup N_n \end{aligned}$$

Then we have a sequence of approximants

$$\mathcal{T}_n^v(x, y) = \frac{N_n(x, y)}{D(x, y)}, \quad n \geq 0$$

where the denominator is fixed (obtained by solving the $(m-1) \times (m-1)$ system (13) and the numerator coefficients are given by (6).

Figure 1: Index sets for vertical sequences



2. **Horizontal sequences:** we consider now the sets:

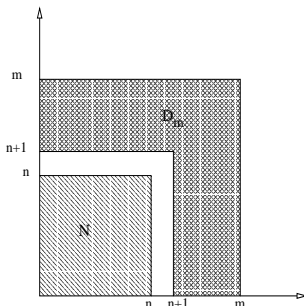
$$\begin{aligned}
 N &= \{(i, j) : 0 \leq i, j \leq n\} \text{ fixed} \\
 D_m &= \{(i, j) : (n+1 \leq i \leq m, 0 \leq j \leq m) \text{ or } (0 \leq i \leq n, n+1 \leq j \leq m)\} \cup \{(0, 0)\} \\
 E_m &= D_m \cup N
 \end{aligned}$$

We have seen that the important part of the computational work to solve the coefficient problem (that is, compute the explicit form of the approximant – its coefficients) is the solution of the system (13) giving the denominator coefficients (the numerator coefficients are then given trivially by (6)). Let us consider the following ordering for the pairs $(i, j) \in \mathbb{N}^2$:

$$(0, 0), (0, 1), (1, 1), (1, 0), (0, 2), (1, 2), (2, 2), (2, 1), (2, 0), \dots$$

If we set $(\mathcal{T}_m^h)_m$ the sequence of Frobenius-Padé approximants defined from the sets of indices (N, D_m, E_m) , then we easily see from (13) that the system we have to solve to compute the denominator coefficients of \mathcal{T}_{m+1}^h is obtained from the previous one (corresponding to \mathcal{T}_m^h) by adding $2m + 1$ rows and columns. So we can solve it efficiently by using the block bordering method [5] and compute efficiently the sequence of approximants.

Figure 2: Index sets for horizontal sequences



3. **Diagonal sequences:** for $m \geq 0$, we define the approximant $\mathcal{T}_m^d(x, y)$ from the following sets:

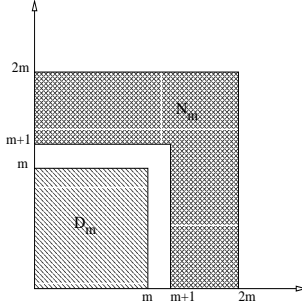
- $D_m = \{(i, j) : 0 \leq i \leq m, 0 \leq j \leq m\}$
- $N_m = \{(i, j) : (m+1 \leq i \leq 2m \text{ and } 0 \leq j \leq 2m) \text{ or } (0 \leq i \leq 2m \text{ and } m+1 \leq j \leq 2m)\} \cup \{(0, 0)\}$
- $E_m = \{(i, j) : 0 \leq i \leq 2m, 0 \leq j \leq 2m\}$

If we denote by \mathbf{b}_m the vector of denominator coefficients at step m and by $\mathbf{H}_m \mathbf{b}_m = \mathbf{0}$ the system to solve, in the next step we will need to solve $\mathbf{H}_{m+1} \mathbf{b}_{m+1} = \mathbf{0}$ with

$$\mathbf{H}_{m+1} = \begin{pmatrix} \mathbf{H}_m & h_{i_2 j_2}^{k_{m'+1} l_{m'+1}} & \dots & h_{i_2 j_2}^{k_{m'+2m+2} l_{m'+2m+2}} \\ & \vdots & & \vdots \\ h_{i_{m'+1} j_{m'+1}}^{k_1 l_1} & \dots & \dots & h_{i_{m'+1} j_{m'+1}}^{k_{m'+2m+2} l_{m'+2m+2}} \\ & \vdots & & \vdots \\ h_{i_{m'+2m+2} j_{m'+2m+2}}^{k_1 l_1} & \dots & \dots & h_{i_{m'+2m+2} j_{m'+2m+2}}^{k_{m'+2m+2} l_{m'+2m+2}} \end{pmatrix}$$

($m' = m^2$). So the system can be obtained from the previous one by adding $2m+1$ new rows and columns and we can then apply, like in the previous case, the bordering block method to solve it efficiently.

Figure 3: Index sets for diagonal sequences



4.2 “Homogeneous” approximants

In this second class of approximants we consider the following choices for the indices sets:

- $E = \{(i, j) : 0 \leq i + j \leq m - 1\}$, $\text{Card}(E) = \frac{m(m+1)}{2}$;
- $N = \{(i, j) : 0 \leq i + j \leq m - l\}$
- $D = \{(i, j) : m - l + 1 \leq i + j \leq m - l\} \cup \{(0, 0)\}$

and we get the approximants

$$\mathcal{H}_{ml}(x, y) = \sum_{i+j=0}^{m-l} a_{ij} P_i(x) P_j(y) / \sum_{i+j=m-l}^{m-1} b_{ij} P_i(x) P_j(y)$$

As in the previous case, we can define sequences in such a way that for computing the $m+1$ th term we can take advantage of the computations done in the previous steps.

For this, let us consider the following ordering in \mathbb{N}^2 :

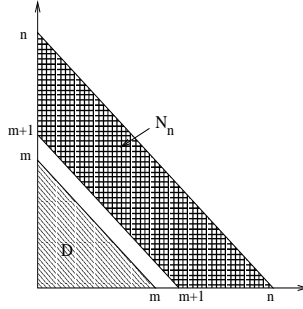
$$(0, 0), (1, 0), (0, 1), (2, 0), (1, 1), (0, 2), \dots, (n, 0), (n-1, 1), \dots, (0, n), \dots$$

- **vertical sequences:** let us consider the sequence of approximants $(\mathcal{H}_m^v)_m$ defined by the indices set

- $D = \{(i, j) : 0 \leq i + j \leq m\}$ for a fixed m ;
- $N_n = \{(i, j) : m + 1 \leq i + j \leq n\} \cup \{(0, 0)\}$
- $E_n = D \cup N_n$

In this case the sequence of approximants has a fixed denominator (obtained by solving a $m(m+1)/2$ square system (13)) and the numerator coefficients follow from (6).

Figure 4: Index sets for vertical sequences



- **horizontal sequences:** let us consider now the sequence $(\mathcal{H}_m^h)_m$ of approximants defined by the indices set:

- $N = \{(i, j) : 0 \leq i + j \leq n\}$ for a fixed n ;
- $D_m = \{(i, j) : n + 1 \leq i + j \leq m\} \cup \{(0, 0)\}$, $m > n + 1$
- $E_m = D_m \cup N$

We easily see that in this case, the system we have to solve to compute \mathcal{H}_{m+1}^h can be obtained from the one corresponding to \mathcal{H}_m^h by adding $m+1$ rows and columns to the system and again we can apply the block bordering method.

- **diagonal sequences:** we will consider a sequence $\mathcal{H}_m^d(x, y)$, $m \geq 0$ defined by the sets

- $D_m = \{(i, j) : 0 \leq i + j \leq m\}$
- $N_m = \{(i, j) : (m + 1 \leq i + j \leq 2m) \cup \{(0, 0)\}$
- $E_m = \{(i, j) : 0 \leq i + j \leq 2m\}$

In the denominator we have $m' = (m + 2)(m + 1)/2$ coefficients, in the numerator $(3m + 3)m/2 + 1$ and the sum gives $\text{Card}(E) - 1$. We have then an homogeneous system of $m' - 1$ equations and m' unknowns which always gives a non trivial solution for the b_{ij} and the numerator coefficients a_{ij} are obtained from

$$d_{ij} = 0 \text{ for } (i, j) = (0, 0) \text{ and } m + 1 \leq i + j \leq 2m$$

If we want to compute the $m + 1$ th approximant after having computed the m th one, then we have to solve a system that, like in the previous section, is obtained from the one of step m by adding a block of $m + 2$ rows and columns. It can then be solved efficiently by the bordering method.

5 Displacement rank for Frobenius-Padé matrices

We will now look for a structure of the matrix of the system giving the denominator coefficients of the approximants, in order to obtain more efficient algorithms to solve the coefficient problem in some particular cases. We recall that to obtain the numerator and denominator coefficients of an approximant associated to the indices sets N, D and E we have to solve the two systems (6) and (5). If we enumerate the elements in these sets by

$$\begin{cases} D = \{(k_1, l_1), (k_2, l_2), \dots, (k_{n+1}, l_{n+1})\} \\ E \setminus N = \{(i_1, j_1), (i_2, j_2), \dots, (i_n, j_n)\} \end{cases}$$

and we set $b_{k_{n+1}l_{n+1}} = 1$ (we recall that we have an homogeneous system of $n + 1$ unknowns and n equations so we fix one unknown), the denominator coefficient matrix is

$$M = \begin{pmatrix} h_{i_1 j_1}^{k_1 l_1} & h_{i_1 j_1}^{k_2 l_2} & \dots & h_{i_1 j_1}^{k_n l_n} \\ h_{i_2 j_2}^{k_1 l_1} & h_{i_2 j_2}^{k_2 l_2} & \dots & h_{i_2 j_2}^{k_n l_n} \\ \dots & \dots & \dots & \dots \\ h_{i_n j_n}^{k_1 l_1} & h_{i_n j_n}^{k_2 l_2} & \dots & h_{i_n j_n}^{k_n l_n} \end{pmatrix} \quad (14)$$

We will look for a displacement structure for M . We recall that M has a $\{Y, V\}$ -displacement structure if

$$VM - MY = GB \text{ with}$$

- $G \in \mathcal{M}_{n \times \alpha}, B \in \mathcal{M}_{\alpha \times n}$ are the generators;
- $\text{rank}(VM - MY) = \alpha$ small compared with the matrix size n .

Fast algorithms to solve $Mx = c$ have been proposed for several types of matrices V and Y (see for instance [18] and the references inside). Gauss elimination applied to M requires $\mathcal{O}(n^3)$ operations. Displacement structure allows to speed up Gauss elimination. In fact the n^2 entries of M are completely determined by the entries of the generators $\{G, B\}$. Translating the Gauss elimination procedure into appropriate operations on the generators gives fast algorithms. Let us now return to our problem and consider the particular case

$$D = \{(i, j) : 0 \leq i, j \leq m - 1\} \text{ and } E \setminus N = D \setminus \{(k_{n+1}, l_{n+1})\}$$

and use the following ordering for this set:

$$(0, 0), (0, 1), (0, 2), \dots, (0, m-1), (1, 0), (1, 1), \dots, (1, m-1), \dots, (m-1, 0), (m-1, 1), \dots, (m-1, m-1).$$

The recurrence relations (8) and (9) obtained in section 2.1 that enable us to compute recursively the quantities h_{ij}^{kl} , give a linear relation involving three consecutive elements of a row and three consecutive elements of a column of the matrix (14) like in the following scheme

$$\begin{array}{ccccc} & & h_{i-1,j}^{kl} & & \\ & h_{ij}^{k-1,l} & h_{ij}^{kl} & h_{ij}^{k+1,l} & \\ & & h_{i+1,j}^{kl} & & \end{array} \quad \text{and} \quad \begin{array}{ccccc} & & h_{i,j-1}^{kl} & & \\ & h_{ij}^{k,l-1} & h_{ij}^{kl} & h_{ij}^{k,l+1} & \\ & & h_{i,j+1}^{kl} & & \end{array}$$

This suggests that if we left-multiply M by an adequate matrix where we store the coefficients of recurrence relations (8),(9) (basically the coefficients of the recurrence relations for the orthogonal system $\{P_k\}$), we right-multiply M by the same matrix, and then subtract the two we must obtain a matrix with a lot of zero entries. Let us see this in detail.

We divide the matrix M in $m \times m$ blocks each one of size $m \times m$.

$$M = \begin{pmatrix} M_{00} & M_{01} & \cdots & M_{0,m-1} \\ M_{10} & M_{11} & \cdots & M_{1,m-1} \\ \vdots & \vdots & \vdots & \vdots \\ M_{m-1,0} & M_{m-1,1} & \cdots & M_{m-1,m-1} \end{pmatrix} \quad (15)$$

where each M_{ij} is given by

$$M_{ij} = \begin{pmatrix} h_{i0}^{j0} & h_{i0}^{j1} & \cdots & h_{i0}^{j,m-1} \\ h_{i1}^{j0} & h_{i1}^{j1} & \cdots & h_{i1}^{j,m-1} \\ \vdots & \vdots & \vdots & \vdots \\ h_{i,m-1}^{j0} & h_{i,m-1}^{j1} & \cdots & h_{i,m-1}^{j,m-1} \end{pmatrix} \quad \text{for } 0 \leq i, j \leq m-1$$

We set

$$F = \begin{pmatrix} F_0 & 0 & \cdots & \cdots \\ 0 & F_0 & 0 & \cdots \\ & & \ddots & \\ 0 & \cdots & 0 & F_0 \end{pmatrix} \quad \text{with } F_0 = \begin{pmatrix} \beta_0 & \gamma_1 & & & \\ \alpha_0 & \beta_1 & \ddots & & \\ & \alpha_1 & \ddots & \ddots & \\ & & \ddots & \ddots & \gamma_{m-1} \\ & & & \alpha_{m-2} & \beta_{m-1} \end{pmatrix}$$

Let us compute the matrix

$$FM - MF = \begin{pmatrix} F_0 M_{00} - M_{00} F_0 & F_0 M_{01} - M_{01} F_0 & \cdots & F_0 M_{0,m-1} - M_{0,m-1} F_0 \\ \vdots & \vdots & \vdots & \vdots \\ F_0 M_{m-1,0} - M_{m-1,0} F_0 & F_0 M_{m-1,1} - M_{m-1,1} F_0 & \cdots & F_0 M_{m-1,m-1} - M_{m-1,m-1} F_0 \end{pmatrix}$$

Let us compute $R_{00} = F_0 M_{00} - M_{00} F_0$.

- $i, j = 2, \dots, m-1$

$$\begin{cases} e_{i+1}^T F_0 M_{00} e_{j+1} = \alpha_{i-1} h_{0,i-1}^{0,j} + \beta_i h_{0i}^{0j} + \gamma_{i+1} h_{0,i+1}^{0j}, & i, j = 1, \dots, m-2 \\ e_{i+1}^T M_{00} F_0 e_{j+1} = \gamma_j h_{0i}^{0,j-1} + \beta_j h_{0i}^{0j} + \alpha_j h_{0i}^{0,j+1}, & i, j = 1, \dots, m-2 \end{cases}$$

and using relation (9) we immediately get

$$e_i^T (F_0 M_{00} - M_{00} F_0) e_j = 0 \quad i, j = 2, \dots, m-1$$

- **i=1**

– for $j = 1, \dots, m-1$ we get

$$e_1^T R_{00} e_j = \beta_0 h_{00}^{0,j-1} + \gamma_1 h_{01}^{0,j-1} - (\gamma_{j-1} h_{00}^{0,j-2} + \beta_{j-1} h_{00}^{0,j-1} + \alpha_{j-1} h_{00}^{0j}) = 0$$

– for $j = m$ and using (9) we get:

$$e_1^T R_{00} e_m = \beta_0 h_{00}^{0,m-1} + \gamma_1 h_{01}^{0,m-1} - (\gamma_{m-1} h_{00}^{0,m-1} + \beta_{m-1} h_{00}^{0,m-1}) = \alpha_{m-1} h_{00}^{0m}$$

- **i = m**

$$\begin{aligned} e_m^T R_{00} e_{j+1} &= (\alpha_{m-2} h_{0,m-2}^{0j} + \beta_{m-1} h_{0,m-1}^{0j}) - (\gamma_j h_{0,m-1}^{0,j-1} + \beta_j h_{0,m-1}^{0j} + \alpha_j h_{0,m-1}^{0,j+1}) = \\ &= \begin{cases} -\gamma_m h_{0,m}^{0,j-1} & j = 1, \dots, m-1 \\ -\gamma_m + h_{0m}^{0,m-1} + \alpha_{m-1} h_{0,m-1}^{0,m} & j = m-1 \end{cases} \end{aligned}$$

- **j = m**

For $i = 2, \dots, m-1$ and again by using (9) we get

$$\begin{aligned} e_i^T R_{00} e_m &= \alpha_{i-2} h_{0,i-2}^{0,m-1} + \beta_{i-1} h_{0,i-1}^{0,m-1} + \gamma_i h_{0i}^{0,m-1} - (\gamma_{m-1} h_{0,i-1}^{0,m-2} + \beta_{m-1} h_{0,i-1}^{0,m-1} + \alpha_{m-1} h_{0,i-1}^{0m}) + \\ &\quad + \alpha_{m-1} h_{0,i-1}^{0,m} = \alpha_{m-1} h_{0,i-1}^{0m} \end{aligned}$$

- **j = 1**

For $i = 2, \dots, m-1$ we obtain

$$\begin{aligned} - e_i^T F_0 M_{00} e_1 &= \alpha_{i-2} h_{0,i-1}^{00} + \beta_{i-1} h_{0,i}^{00} + \gamma_{i-1} h_{0,i+1}^{00} \\ - e_i^T M_{00} F_0 e_1 &= \beta_0 h_{0,i-1}^{00} + \alpha_0 h_{0,i-1}^{0,1} \end{aligned}$$

and, as $\gamma_0 = 0$, by using (9) we get

$$e_i^T R_{00} e_1 = 0 \quad i = 2, \dots, m-1$$

Let us multiply in the left and in the right the matrix R by two permutation matrices P_1 and P_2 in order to group all non-zero columns on the right hand side and all the non-zero rows in the bottom of the matrix. We obtain

$$P_1 R P_2 = \begin{pmatrix} 0 & \alpha_{m-1} T \\ -\gamma_m S & U \end{pmatrix} \text{ with } T \in \mathcal{M}_{((m-1)m) \times m}, S \in \mathcal{M}_{m \times ((m-1)m)} \text{ and } U \in \mathcal{M}_{m \times m}$$

If we set

$$U_0 = \begin{pmatrix} h_{0,m-1}^{0m} & \cdots & h_{0,m-1}^{m-1,m} \\ \vdots & \vdots & \vdots \\ h_{m-1,m-1}^{0m} & \cdots & h_{m-1,m-1}^{m-1,m} \end{pmatrix}, U_1 = \begin{pmatrix} h_{0,m}^{0,m-1} & \cdots & h_{0,m}^{m-1,m-1} \\ \vdots & \vdots & \vdots \\ h_{m-1,m}^{0,m-1} & \cdots & h_{m-1,m}^{m-1,m-1} \end{pmatrix}$$

then

$$U = \alpha_{m-1} U_0 - \gamma_m U_1$$

If we define the matrices

$$G = \begin{pmatrix} 0 & T \\ -\gamma_m I & U_0 \end{pmatrix} \text{ and } B = \begin{pmatrix} S & U_1 \\ 0 & \alpha_{m-1} I \end{pmatrix} \text{ then}$$

$$GB = \begin{pmatrix} 0 & \alpha_{m-1} T \\ -\gamma_m S & -\gamma_m U_1 + \alpha_{m-1} U_0 \end{pmatrix} = \begin{pmatrix} 0 & \alpha_{m-1} T \\ -\gamma_m S & U \end{pmatrix} = P_1 R P_2$$

So

$$R = G' B' \text{ with } G' = P_1^T G \in \mathcal{M}_{m^2 \times (2m)}, B' = B P_2^T \in \mathcal{M}_{(2m) \times m^2}$$

Finally we get

$$FM - MF = G' B'$$

which means that M has a Hessenberg displacement structure with displacement rank $2m$.

Similar results can be obtain for more general index sets. In fact, we can show the following result:

Theorem 1 *Let us consider the multivariate Frobenius-Padé approximant for*

$$f(x, y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} f_{ij} P_i(x) P_j(y)$$

defined from the index sets

$$D = [i', i' + m - 1] \times [j', j' + m - 1] \text{ and } E \setminus N = [i^*, i^* + m - 1] \times [j^*, j^* + m - 1]$$

and let M be the $m^2 \times m^2$ matrix of the system to solve to obtain the denominator coefficients. Then M has an Hessenberg displacement rank structure with rank $4m$.

Proof:

the proof involves rather long computations using the same techniques as in the previous case. We can summarize these computations as follows.

Let M be the denominator matrix of size $m^2 \times m^2$ divides into m^2 square blocks of size m like in (15). We define the tridiagonal matrix of size $m \times m$

$$F_j = \begin{pmatrix} \beta_j & \gamma_{j+1} & & & \\ \alpha_j & \beta_{j+1} & \cdots & & \\ & \alpha_{j+1} & \cdots & \cdots & \\ & & \cdots & \cdots & \gamma_{j+m-1} \\ & & & \alpha_{j+m-2} & \beta_{j+m-1} \end{pmatrix}.$$

Using relations (9) we obtain

$$F_{j^*} M_{ij} - M_{ij} F_{j'} = \begin{pmatrix} \times & \times & \cdots & \times & \times \\ \times & 0 & \cdots & 0 & \times \\ \vdots & \vdots & & \vdots & \vdots \\ \times & 0 & \cdots & 0 & \times \\ \times & \times & \cdots & \times & \times \end{pmatrix},$$

a matrix with nonzero entries only in the first and last rows and columns. If we define

$$\mathcal{F}_{j^*} = \text{diag}(F_{j^*}) \in \mathcal{M}_{m^2 \times m^2}, \mathcal{F}_{j'} = \text{diag}(F_{j'}) \in \mathcal{M}_{m^2 \times m^2}$$

it can be shown that

$$P(\mathcal{F}_{j^*} M - M \mathcal{F}_{j'}) Q = \begin{pmatrix} 0 & T \\ S & U \end{pmatrix} \text{ where}$$

- P, Q are permutation matrices;
- $T = T' \times Z'_{j'}$, $T' \in \mathcal{M}_{(m(m-2)) \times (2m)}$,
 $Z'_{j'} = \text{diag}(\gamma_{j'}, \cdots, \gamma_{j'}, \alpha_{j'+m-1}, \cdots, \alpha_{j'+m-1}) \in \mathcal{M}_{2m \times 2m}$;
- $S = -Z_{j^*}^* \times S'$, $S' \in \mathcal{M}_{(2m) \times (m(m-2))}$,
 $Z_{j^*}^* = \text{diag}(\alpha_{j^*-1}, \cdots, \alpha_{j^*-1}, \gamma_{j^*+m}, \cdots, \gamma_{j^*+m}) \in \mathcal{M}_{2m \times 2m}$
- $U = U_0 Z'_{j'} - Z_{j^*}^* U_1 \in \mathcal{M}_{(2m) \times (2m)}$
- U_0, U_1, S', T' are matrices with entries h_{ij}^{kl} .

Finally if we define

$$G = \begin{pmatrix} 0 & T' \\ -Z_{j^*}^* & U_0 \end{pmatrix} \in \mathcal{M}_{(4m) \times m^2} \text{ and } B = \begin{pmatrix} S' & U_1 \\ 0 & -Z'_{j'} \end{pmatrix} \in \mathcal{M}_{(m^2) \times 4m}$$

we easily obtain

$$P(\mathcal{F}_{j^*} M - M \mathcal{F}_{j'}) Q = GB$$

which gives for M a Hessenberg displacement structure with displacement rank $4m$.

We conclude that for the general class of multivariate Frobenius-Padé approximants corresponding to square index sets D and $E \setminus N$ the denominator coefficient matrix M is a polynomial Hankel like matrix [20] because \mathcal{F}_{j^*} and $\mathcal{F}_{j'}$ are Hessenberg (more precisely, tridiagonal).

Fast algorithms (Levinson type algorithms, that is, that produce a triangular factorization of the inverse of the matrix) for solving systems with Hessenberg displacement structure have been proposed in [15]. In [17] a Schur type algorithm has been proposed for recursive triangular factorization of general polynomial Hankel-like matrices, that is, matrices with a $\{Y, V\}$ -displacement structure defined via Hessenberg matrices V and Y . That algorithm can be applied to our matrices leading to a fast solution of the system. In fact, its complexity is given in [17]:

$$C(n) = \mathcal{O}(M(n)n + n^2)$$

for a $n \times n$ matrix, where $M(n)$ is the cost of a multiplication of Y, V by a vector. In our case the complexity will be of order m^4 instead of m^6 . The development of such algorithms with application to the computation of multivariate Frobenius Padé approximants is under study.

6 Mixed Frobenius-Padé approximants

6.1 Definitions

We will now generalize to the Frobenius-Padé case the definition of nested multivariate Padé approximants given in [13] for two variable functions given by a power series expansion. This approach consists in applying the Padé approximation with respect to y to the coefficients of the Padé approximation with respect to x . This leads to algorithms involving univariate approximation (and so small systems) and faster in terms of computational costs (FFT methods can be used) when compared to other generalizations of Padé approximants to multivariate case. Convergence results for these approximants were given in [14].

As in the previous sections we suppose that $f(x, y)$ is a function given by its expansion in an orthogonal system $\{P_k\}$ and we write it in the following way:

$$f(x, y) = \sum_{j=0}^{\infty} \sum_{i=0}^{\infty} f_{ij} P_i(y) P_j(x) = \sum_{j=0}^{\infty} f_j(y) P_j(x).$$

The approximant is constructed in two steps.

First step: we consider f as a function of x (and y will be considered as a parameter)

$$f_y(x) = \sum_{j=0}^{\infty} f_j(y) P_j(x).$$

We want to compute two polynomials $Q_y(x)$ and $P_y(x)$ of the form

$$\begin{cases} Q_y(x) &= 1 + \sum_{i=1}^m b_i(y) x^i \\ P_y(x) &= \sum_{i=0}^n a_i(y) P_i(x) \end{cases}$$

such that:

$$Q_y(x)f_y(x) - P_y(x) = \mathcal{O}(P_{n+m+1}(x)),$$

which means that the first $n + m + 1$ coefficients in the expansion in the orthogonal system $\{P_k\}$ vanish. This is a kind of Frobenius-Padé approximant for the one variable function $f_y(x)$: the only difference is that we express the denominator in the powers of x and not in the orthogonal system $\{P_k\}$ in order to simplify the computations.

Replacing the expressions of $Q_y(x)$ and $P_y(x)$ we get:

$$\sum_{j=0}^{\infty} f_j(y)P_j(x) + \sum_{j=0}^{\infty} \sum_{i=1}^m b_i(y)f_j(y)x^i P_j(x) - \sum_{i=0}^n a_i(y)P_i(x) = \mathcal{O}(P_{n+m+1}(x)).$$

We set

$$x^i \sum_{j=0}^{\infty} f_j(y)P_j(x) = \sum_{j=0}^{\infty} f_j^i(y)P_j(x), \quad (16)$$

and finally

$$\sum_{j=0}^{\infty} \left(f_j(y) + \sum_{i=1}^m b_i(y)f_j^i(y) \right) P_j(x) - \sum_{i=0}^n a_i(y)P_i(x) = \mathcal{O}(P_{n+m+1}(x)).$$

This leads to the following system:

$$\sum_{i=1}^m b_i(y)f_j^i(y) + f_j(y) = 0 \quad j = n+1, \dots, n+m \quad (17)$$

$$\sum_{i=1}^m b_i(y)f_j^i(y) + f_j(y) = a_j(y) \quad j = 0, \dots, n \quad (18)$$

Second step: we replace the unknown functions $a_j(y)$, $j = 0, \dots, n$ and $b_i(y)$, $i = 0, \dots, m$ (solution of our problem) by some polynomials that approach them in the following way

- (a) for $i = 1, \dots, m$

$$b_i(y) = \sum_{k=0}^M b_{ik}y^k.$$

We will have $m(M+1)$ coefficients to compute and we can choose them in order to annihilate the first $M+1$ coefficients of the expansion in the orthogonal system $\{P_k\}$ of each equation in (17). We need to obtain the expansion in the orthogonal system $\{P_k\}$ of the left hand side of (17), that is for the functions $f_j^i(y)$. We obtain the following result:

Proposition 1 *Let $f(x, y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} f_{ij}P_i(x)P_j(y)$ be a two variable function given by its expansion in an orthogonal system $\{P_k\}$ satisfying the recurrence relation*

$$xP_k(x) = A_kP_{k+1}(x) + B_kP_k(x) + C_kP_{k-1}(x), \quad (19)$$

and let us set as previously,

$$f_y(x) = \sum_{j=0}^{\infty} f_j(y) P_j(x), \quad f_j(y) = \sum_{k=0}^{\infty} f_{jk} P_k(y).$$

1. The polynomials $x^i P_k(x)$ can be expressed in the $\{P_k\}$ basis in the following way

$$x^i P_k(x) = \alpha_{k,k+i}^i P_{k+i}(x) + \cdots + \alpha_{kk}^i P_k(x) + \cdots + \alpha_{k,k-i}^i P_{k-i}(x), \quad (20)$$

where the coefficients α_{kj}^i can be computed by a recurrence on the upper index i in the following way:

$$\begin{cases} \alpha_{k,k+i}^i &= A_{k+i-1} \alpha_{k,k+i-1}^{i-1} \\ \alpha_{k,k-i}^i &= C_{k-i+1} \alpha_{k,k-i+1}^{i-1} \\ \alpha_{kj}^i &= A_{j-1} \alpha_{k,j-1}^{i-1} + B_j \alpha_{kj}^{i-1} + C_{j+1} \alpha_{k,j+1}^{i-1}, \quad k-i+1 \leq j \leq k+i-1 \end{cases}. \quad (21)$$

2. Let us consider the expansion of $x^i f_y(x)$ in the orthogonal system $\{P_k\}$,

$$x^i f_y(x) = \sum_{j=0}^{\infty} f_j^i(y) P_j(x).$$

Then the expansion of the functions $f_j^i(y)$ in the system $\{P_k\}$ is given by

$$f_j^i(y) = \sum_{k=0}^{\infty} f_{jk}^i P_k(y) \quad \text{with} \quad f_{jk}^i = \sum_{n=k-i}^{k+i} f_{nk} \alpha_{nj}^i$$

Proof:

1. this can be easily proved by induction using (19).

2. we have $x^i f_y(x) = \sum_{j=0}^{\infty} f_j(y) x^i P_j(x)$. From (20) we see that the contribution to the coefficient of $P_k(x)$ comes from the terms $x^i P_{k-i}(x), \dots, x^i P_{k+i}(x)$ and so

$$\begin{aligned} f_k^i(y) &= \sum_{j=k-i}^{k+i} f_j(y) \alpha_{j,k}^i = \sum_{j=k-i}^{k+i} \left(\sum_{l \geq 0} f_{jl} P_l(y) \right) \alpha_{j,k}^i = \\ &= \sum_{l \geq 0} \left(\sum_{j=k-i}^{k+i} f_{jl} \alpha_{jk}^i \right) P_l(y) \end{aligned}$$

and the result follows.

This proposition enables us to obtain the system to be solved in order to compute the coefficients b_{kl} of the denominator polynomial. It is obtained from

$$\begin{pmatrix} f_{n+1}^1(y) & f_{n+1}^2(y) & \cdots & f_{n+1}^m(y) \\ f_{n+2}^1(y) & f_{n+2}^2(y) & \cdots & f_{n+2}^m(y) \\ \vdots & \vdots & \cdots & \vdots \\ f_{n+m}^1(y) & f_{n+m}^2(y) & \cdots & f_{n+m}^m(y) \end{pmatrix} \begin{pmatrix} b_1(y) \\ b_2(y) \\ \vdots \\ b_m(y) \end{pmatrix} = - \begin{pmatrix} f_{n+1}(y) \\ f_{n+2}(y) \\ \vdots \\ f_{n+m}(y) \end{pmatrix}$$

equating the coefficients of the first $M + 1$ terms of each equation.

- (b) for $j = 0, \dots, n$

$$a_j(y) = \sum_{k=0}^N a_{jk} P_k(y)$$

and so we will have to compute $(n+1)(N+1)$ coefficients (a_{jk}) that will be chosen in order to equalize the first $N + 1$ coefficients of the expansion in $\{P_k\}$ of the right and left hand side of each equation of (18). The expansion of the left hand side in the orthogonal system $\{P_k\}$ is obtained as in (a).

(a) implies the resolution of a linear system of $m(M + 1)$ equations and from (b) we obtain directly the coefficients a_{ij} . We will call **Mixed Frobenius-Padé approximant** the rational approximant

$$R(x, y) = \frac{P(x, y)}{Q(x, y)} \text{ with } \begin{cases} P(x, y) = \sum_{i=0}^n \sum_{k=0}^N a_{ik} P_i(x) P_k(y) \\ Q(x, y) = 1 + \sum_{i=1}^m \sum_{k=0}^M b_{ik} x^i y^k \end{cases}$$

satisfying

$$Q(x, y) f(x, y) - P(x, y) = \sum_{(i,j) \in (\mathbb{N}^2 \setminus E)} d_{ij} P_i(x) P_j(y)$$

with

$$E = \{(i, j) : (0 \leq i \leq n \text{ and } 0 \leq j \leq N) \text{ or } (n+1 \leq i \leq n+m \text{ and } 0 \leq j \leq M)\}$$

The shape and the cardinal of the set of indices E depends on the number of the known coefficients of the series $f(x, y)$. If we want a symmetry with respect to the two variables x and y it is sufficient to take $N = M = n + m$. But in this case we don't have symmetry in the approximant - the same powers of x and y . For this we need to take $N = n$ and $M = m$, and we will privilege the x variable. This can be a very interesting property of these approximants: we can expect to obtain good approximations when the function has an unsymmetric set of singularities because of the different nature of the variables

We remark that we can do the same type of construction beginning with the y variable.

As we have seen, in the case of a general family of orthogonal polynomials we need a large amount of computations to construct the entries of the linear system that give the coefficients of the approximant, and it seems difficult to obtain a structure for this matrix. We will now consider a particular case - the family of Tchebychev polynomials - for which the previous computations will be very simple allowing us to obtain a structure for the coefficient matrix.

6.2 A class of Padé-Tchebyshev approximants in two variables

Let us now consider a function $f(x, y)$ given by

$$f(x, y) = \sum_{i,j \geq 0}^{\infty} f_{ij} T_i(x) T_j(y) = \sum_{j=0}^{\infty} f_j(y) T_j(x)$$

where $\{T_k\}$ is the system of Tchebyshev polynomials of first kind.

We will proceed as in the previous section to construct the **mixed Padé-Tchebyshev approximant** of $f(x, y)$ following the two steps.

First step: we construct a linearized Padé-Tchebyshev approximant to the function of x , $f_y(x) = f(x, y)$, that is, we construct the two polynomials $Q_y(x)$ and $P_y(x)$ of the form:

$$\begin{cases} Q_y(x) &= \sum_{i=0}^m b_i(y) T_i(x) \\ P_y(x) &= \sum_{i=0}^n a_i(y) T_i(x) \end{cases}$$

such that

$$Q_y(x) f_y(x) - P_y(x) = \mathcal{O}(T_{n+m+1}(x)).$$

We recall a fundamental property of the Tchebyshev polynomials:

$$T_k(x) T_i(x) = \frac{1}{2} (T_{k+i}(x) + T_{k-i}(x)) \quad i \geq 0, k \geq 0 \quad (22)$$

with the notation $T_{-i}(x) = T_i(x)$ for all $i \geq 0$.

Using (22) we get:

$$\begin{aligned} & \left(\sum_{k=0}^m b_k(y) T_k(x) \right) \left(\sum_{i=0}^{\infty} f_i(y) T_i(x) \right) = \\ &= \frac{1}{2} \sum_{k=0}^m b_k(y) \left(\sum_{i=k}^n f_{i-k}(y) T_i(x) + \sum_{i=-k}^n f_{i+k}(y) T_i(x) + \sum_{i=n+1}^{\infty} (f_{i-k}(y) + f_{i+k}(y)) T_i(x) \right) = \\ &= \frac{1}{2} \sum_{i=n+1}^{\infty} \left(\sum_{k=0}^m b_k(y) (f_{i-k}(y) + f_{i+k}(y)) \right) T_i(x) + \\ & \quad + \frac{1}{2} \sum_{k=0}^m b_k(y) \left[\sum_{i=0}^n f_{i-k}(y) T_i(x) + \sum_{i=1}^n f_{i+k}(y) T_i(x) + f_0(y) T_k(x) \right] = \\ &= \frac{1}{2} \sum_{i=n+1}^{\infty} \left(\sum_{k=0}^m b_k(y) (f_{i-k}(y) + f_{i+k}(y)) \right) T_i(x) + \\ & \quad + \frac{1}{2} \sum_{i=1}^n \left(\sum_{k=0}^m b_k(y) (f_{i-k}(y) + f_{i+k}(y)) \right) T_i(x) + \frac{1}{2} f_0(y) \sum_{k=0}^m b_k(y) T_k(x) + \frac{1}{2} \sum_{k=0}^m b_k(y) f_k(y) T_0(x) \end{aligned}$$

Let us suppose that $m \leq n$. Then

- the $b_k(y)$ are solution of the system:

$$\sum_{k=0}^m b_k(y) (f_{i-k}(y) + f_{i+k}(y)) = 0, \quad i = n+1, \dots, n+m. \quad (23)$$

We consider the Tchebyshev expansion of the functions $b_j(y)$,

$$b_j(y) = \sum_{i=0}^{\infty} b_{ji} T_i(y), \quad j = 1, \dots, m$$

and we fix $b_0(y) = 1$. We define

$$B(y) = \begin{pmatrix} b_1(y) \\ \vdots \\ b_m(y) \end{pmatrix} = \sum_{i=0}^{\infty} B_i T_i(y) \text{ with } B_i = \begin{pmatrix} b_{1i} \\ b_{2i} \\ \vdots \\ b_{mi} \end{pmatrix}, \quad (24)$$

and we denote by $H^m(y)$ the following $m \times m$ matrix

$$H^m(y) = \begin{pmatrix} f_n(y) + f_{n+2}(y) & f_{n-1}(y) + f_{n+3}(y) & \cdots & f_{n+1-m}(y) + f_{n+1+m}(y) \\ \cdots & \cdots & \cdots & \cdots \\ f_{n+m-1}(y) + f_{n+m+1}(y) & f_{n+m-2}(y) + f_{n+m+2}(y) & \cdots & f_n(y) + f_{n+2m}(y) \end{pmatrix}.$$

Then the $b_j(y)$ are the solution of the system

$$H^m(y)B(y) = F(y) \text{ with } F(y)^T = 2 \begin{pmatrix} f_{n+1}(y) & \cdots & f_{n+m}(y) \end{pmatrix}. \quad (25)$$

- the $a_k(y)$ are given by:

$$\begin{cases} a_0(y) &= \frac{1}{2} \sum_{k=0}^m b_k(y) f_k(y) \\ a_i(y) &= \frac{1}{2} [f_0(y) b_i(y) + \sum_{k=0}^m b_k(y) (f_{i+k}(y) + f_{i-k}(y))] \end{cases} \quad i = 1, \dots, n. \quad (26)$$

Second step: we replace the fonctions $a_i(y)$ and $b_i(y)$ by their approximation by polynomials in the Tchebyshev basis, $a_i^*(y)$ and $b_i^*(y)$.

Computation of the $b_j^*(y)$.

In order to obtain the coefficients b_{ij} we are going to replace B by a polynomial B^* of degree m (that is, truncate the expansion in (24) at order m),

$$B^*(y) = \begin{pmatrix} b_1^*(y) \\ \vdots \\ b_m^*(y) \end{pmatrix} = \sum_{i=0}^m B_i^* T_i(y).$$

The expansion of the right hand side of (25) in the Tchebyshev system becomes

$$\begin{aligned}
H^m(y)B^*(y) &= \\
&= \left(\sum_{i=0}^{\infty} H_i T_i(y) \right) \left(\sum_{j=0}^m B_j^* T_j(y) \right) = \sum_{i=0}^{\infty} \left(\sum_{j=0}^m H_i B_j^* T_i(y) T_j(y) \right) = \\
&= \frac{1}{2} \left(\sum_{i=0}^{\infty} \sum_{j=0}^m H_i B_j^* T_{i+j}(y) + \sum_{i=0}^{\infty} \sum_{j=0}^m H_i^m B_j^* T_{i-j}(y) \right),
\end{aligned}$$

where, for all $i \geq 0$, $H_i \in \mathcal{M}_{m \times m}$ is the i -th coefficient of the expansion of $H^m(y)$ in Tchebyshev series.

- $\sum_{i=0}^{\infty} \sum_{j=0}^m H_i^m B_j^* T_{i+j}(y) = \sum_{j=0}^m \sum_{k=j}^{\infty} H_{k-j} B_j^* T_k(y) =$

$$\sum_{k=0}^{\infty} \left(\sum_{j=0}^k H_{k-j} B_j^* \right) T_k(y)$$
- $\sum_{i=0}^{\infty} \sum_{j=0}^m H_i B_j^* T_{i-j}(y) = \sum_{j=0}^m \sum_{k=-j}^{\infty} H_{k+j} B_j^* T_k(y) =$

$$= \sum_{j=0}^m \sum_{k=-j}^{-1} H_{k+j} B_j^* T_k(y) + \sum_{j=0}^m \sum_{k=0}^{\infty} H_{k+j} B_j^* T_k(y) =$$

$$= \sum_{j=0}^m \sum_{l=1}^j H_{j-l} B_j^* T_l(y) + \sum_{k=0}^{\infty} \left(\sum_{j=0}^m H_{k+j} B_j^* \right) T_k(y) =$$

$$= \sum_{l=1}^m \sum_{j=l}^m H_{j-l} B_j^* T_l(y) + \sum_{k=0}^{\infty} \left(\sum_{j=0}^m H_{k+j} B_j^* \right) T_k(y) =$$

$$= \sum_{k=1}^m \left(\sum_{j=k}^m H_{j-k} B_j^* \right) T_k(y) + \sum_{k=0}^{\infty} \left(\sum_{j=0}^m H_{k+j} B_j^* \right) T_k(y)$$

Regrouping the coefficients corresponding to the same index k in the expansion in $\{T_k\}$ and replacing H_0 and F_0 by $\frac{1}{2}H_0$ and $\frac{1}{2}F_0$ respectively, we finally obtain the linear system:

$$\begin{cases} 2 \sum_{j=0}^m H_j B_j^* = F_0 \\ \sum_{j=0}^m H_{j-k} B_j^* + \sum_{j=0}^m H_{j+k} B_j^* = F_k, \quad k = 1, \dots, m \end{cases}$$

(with the convention $H_i = H_{-i}$), which corresponds to equating the coefficients of terms up to order m in both sides of (25).

We have then shown the following result:

Theorem 2 *Let f be a two variable function given by its expansion in a Tchebyshev series*

$$f(x, y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} f_{ij} T_i(x) T_j(y) = \sum_{j=0}^{\infty} f_j(y) T_j(x)$$

We define **mixed Padé-Tchebyshev approximant** by

$$PTche_{m,n}(x, y) = P(x, y)/Q(x, y) \text{ with } \begin{cases} P(x, y) = \sum_{j=0}^m b_j^*(y) T_j(x) \text{ with } b_j^*(y) = \sum_{k=0}^m b_{jk}^* T_k(y) \\ Q(x, y) = \sum_{i=0}^n a_i^*(y) T_i(x) \text{ with } a_i^*(y) = \sum_{k=0}^m a_{ik}^* T_k(y) \end{cases}$$

and the coefficients defined by:

- the denominator coefficients b_{ij}^* , regrouped in a vector $\mathcal{B} = (B_0^*, B_1^*, \dots, B_m^*)^T$, $B_j^* = (b_{j1}^*, \dots, b_{jm}^*)^T$, $j = 0, \dots, m$, are the solution of the $(m(m+1)) \times (m(m+1))$ system:

$$\sum_{j=0}^m (H_{j-k} + H_{j+k}) B_j^* = F_k \quad k = 0, \dots, m \quad \Leftrightarrow \quad \mathcal{H}\mathcal{B} = \mathcal{F} \quad (27)$$

with

$$\mathcal{H} = \begin{pmatrix} H_0 & H_1 & \cdots & H_m \\ H_{-1} & H_0 & \cdots & H_{m-1} \\ \vdots & \vdots & & \vdots \\ H_{-m} & H_{-m+1} & \cdots & H_0 \end{pmatrix} + \begin{pmatrix} H_0 & H_1 & \cdots & H_m \\ H_1 & H_2 & \cdots & H_{m+1} \\ \vdots & \vdots & & \vdots \\ H_m & H_{m+1} & \cdots & H_{2m} \end{pmatrix} = \mathcal{H}_1 + \mathcal{H}_2$$

with \mathcal{H}_1 a block Toeplitz matrix and \mathcal{H}_2 a block Hankel matrix;

- the numerator is given by

$$\begin{cases} a_0^*(y) &= \frac{1}{2} \sum_{k=0}^m b_k^*(y) f_k^{(m)}(y) \\ a_i^*(y) &= \frac{1}{2} \left[f_0^{(m)}(y) b_i^*(y) + \sum_{k=0}^m b_k^*(y) \left(f_{i+k}^{(m)}(y) + f_{i-k}^{(m)}(y) \right) \right] \quad i = 1, \dots, n \end{cases}$$

with $f_j^{(m)}(y) = \sum_{k=0}^m f_{jk} T_k(y)$ the partial sums of order m .

Then the polynomials $P(x, y)$ and $Q(x, y)$ satisfy:

$$Q(x, y)f(x, y) - P(x, y) = R(x, y) = \sum_{i,j \geq 0} r_{ij} T_i(x) T_j(y)$$

with

$$r_{ij} = 0 \text{ for } i = 0, \dots, n+m \text{ and } j = 0, \dots, m.$$

This proposition enables us to propose fast algorithms to compute the denominator coefficients. These will be based on the fast inversion algorithms of Toeplitz-plus-Hankel matrices developed in [16]. The formulae and algorithms presented there can be generalized to the block case, that is, $A = T + H$, where T is block Toeplitz and H is block Hankel, and it is easy to show that the solution of the $[m(m+1)] \times [m(m+1)]$ system (27) can be achieved with a complexity of $\mathcal{O}(m^2(m+1)^2)$ (instead of $\mathcal{O}(m^6)$ using Gauss elimination). Implementation of these algorithms and numerical examples will be presented in a future work.

6.3 Padé-Tchebyshev approximants of "tensor product type"

Let us now give a second definition for Padé-Tchebyshev approximants which we call **tensor type** because the denominator polynomial is chosen to be a tensor product of a polynomial of degree m in x by a polynomial of degree m in y . Let us consider the polynomials

$$\begin{cases} Q_m(x, y) &= \sum_{i=0}^m \sum_{j=0}^m b_{ij} T_i(x) T_j(y) \\ P_n(x, y) &= \sum_{(i,j) \in N} a_{ij} T_i(x) T_j(y) \text{ with } N \subset \mathbb{N}^2 \end{cases}$$

We begin by obtaining the expansion of $Q_m f$ in a Tchebyshev series. After some simple but rather long computations using the property (22) we can obtain:

$$\begin{aligned}
Q_m(x, y)f(x, y) &= \left(\sum_{i=0}^m \sum_{j=0}^m b_{ij} T_i(x) T_j(y) \right) \left(\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} f_{ij} T_i(x) T_j(y) \right) = \\
&= \sum_{r=0}^{\infty} \sum_{s=0}^{\infty} T_r(x) T_s(y) \left(\sum_{i=0}^m \sum_{j=0}^m b_{ij} f_{r-i, j-s} \right) + \sum_{r=0}^{\infty} \sum_{s=1}^{\infty} T_r(x) T_s(y) \left(\sum_{i=0}^m \sum_{j=0}^m b_{ij} f_{r-i, j+s} \right) + \\
&+ \sum_{r=1}^{\infty} \sum_{s=0}^{\infty} T_r(x) T_s(y) \left(\sum_{i=0}^m \sum_{j=0}^m b_{ij} f_{r+i, s-j} \right) + \sum_{r=1}^{\infty} \sum_{s=1}^{\infty} T_r(x) T_s(y) \left(\sum_{i=0}^m \sum_{j=0}^m b_{ij} f_{r+i, j+s} \right) \\
&= \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} e_{ij} T_i(x) T_j(y)
\end{aligned} \tag{28}$$

We want to compute a rational approximant $P_n(x, y)/Q_m(x, y)$ to $f(x, y)$ in such a way that:

$$R_m(x, y) = Q_m(x, y)f(x, y) - P_n(x, y) = \sum_{(i,j) \in \mathbb{N}^2 \setminus E} r_{ij} T_i(x) T_j(y)$$

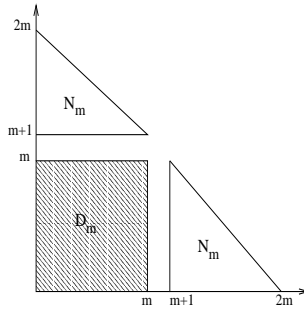
where E is a set of pairs (i, j) for which the corresponding coefficients in the error term are 0. We write $E = N \cup D$, where N and D are disjoint sets, and, as before, N corresponds to the set of indices appearing in the numerator polynomial and D to the set of indices appearing in the denominator. Different types of approximants can be obtained by choosing different sets. In our case we have

$$D = \{(i, j) : 0 \leq i \leq m, 0 \leq j \leq m\}$$

and we will consider the particular choice for N :

$$N = \{(i, j) : m+1 \leq i \leq 2m, 0 \leq j \leq m-i\} \cup \{(i, j) : m+1 \leq j \leq 2m, 0 \leq i \leq m-j\}$$

Figure 5: Numerator and denominator index sets



We will compute the coefficients of the approximant in the following way:

- we begin by computing the coefficients of the denominator (b_{ij}) by

$$e_{ij} = 0 \text{ for } (i, j) \in D, \quad e_{ij} = 0 \quad i = 0, \dots, m, \quad j = 0, \dots, m$$

This corresponds to solving a linear system: with the convention $f_{-i,-j} = f_{ij}$ and setting H_{ij} the following block of size $(m+1)$,

$$H_{ij} = \begin{pmatrix} f_{ij} & f_{i+1,j} & \cdots & f_{i+m,j} \\ f_{i,j+1} & f_{i+1,j+1} & \cdots & f_{i+m,j+1} \\ \vdots & \vdots & & \vdots \\ f_{i,j+m} & f_{i+1,j+m} & \cdots & f_{i+m,j+m} \end{pmatrix},$$

from (28) we immediately obtain the coefficient matrix \mathcal{H} of the system we have to solve to get the (b_{ij}) :

$$\mathcal{H} = (\mathcal{H}_{ij})_{i,j=0}^m, \quad \text{with } \mathcal{H}_{ij} = H_{ij} + H_{-i,j} + H_{i,-j} + H_{-i,-j}.$$

This matrix is easier to construct than in the general case considered in paragraph **6.1**, but not so simple that in the previous case (section **6.2**) where we were able to give for the matrix a structure of block Toeplitz-plus-Hankel.

- the coefficients (a_{ij}) are immediately obtained from the (b_{ij}) by setting

$$a_{ij} = e_{ij} \text{ for } (i, j) \in N$$

(as seen before the e_{ij} are computed from the coefficients of the series and from the b_{ij})

So the principal part of the computational effort corresponds to the computation of the denominator coefficients.

7 Conclusions and future work.

We have proposed different types of multivariate rational approximants for functions given by their orthogonal expansions, based on different generalizations of the concept of Padé approximation to multivariate series. We were mainly interested in proposing algorithms for their computation: recursive algorithms for the computation of the values at a given point of particular sequences of approximants and fast algorithms based on a displacement structure of the matrix giving the denominator coefficients. We are now going to implement these algorithms in order to show their good numerical properties.

The convergence properties of these approximants are under study. In the case of one variable, the Frobenius-Padé approximants gave very good numerical results and a great acceleration of convergence has been shown in [22] for some classes of functions, particularly near the singularities of these functions. So we can expect that these good properties will be generalized to the multivariate case. We will try to obtain acceleration results, that is, from some properties

on the function f or of the sequences (f_{ij}) , we will show that the sequence of multivariate Frobenius-Padé approximants converge faster than the corresponding sequences of partial sums, $S_N(x, y) = \sum_{(i,j) \in N} f_{ij} P_i(x) P_j(y)$.

This will be the object of a forthcoming paper.

References

- [1] G. A. Baker, Jr., P. Graves-Morris, Padé Approximants, Encyclopedia of Mathematics and its Applications, Cambridge University Press 1996.
- [2] C. Brezinski, Padé Approximation and General Orthogonal polynomials, ISNM, vol. 50, Birkhauser-Verlag, Basel, 1980.
- [3] C. Brezinski, *A general Extrapolation Algorithm*, Numer. Math., 35 (1980) 175-187.
- [4] C. Brezinski, *The Mulhbach-Neville-Aitken algorithm and some extensions*, BIT 20, 444-451, 1980.
- [5] C. Brezinski, M. Redivo Zaglia, Extrapolation Methods. Theory and Practice. North-Holland, Amsterdam, 1991.
- [6] J.S. Chisholm, *Rational Approximants defined from double power series*, Math. of Comput. 27 (1973) 841-848.
- [7] A. Cuyt, *Multivariate Padé approximants revisited*, BIT 26 (1986) 71-79.
- [8] A. Cuyt, *Multivariate Padé approximants*, Journ. Math. Anal. Appl. 96, 283-293 (1983).
- [9] A. Cuyt, *A comparison of some multivariate Padé-approximants*, SIAM Journ. Math. Anal. , Vol.14, 195-202 (1983).
- [10] A. Cuyt, *A review of multivariate Padé Approximation theory*, J. Comput. Appl. Math. 12 (1985), 221-232.
- [11] A. Cuyt, *How well can the concept of Padé approximant be generalized to the multivariate case?* , J. Comput. Appl. Math. 105 (1999) 25-50.
- [12] V.N. Faddeeva, Computational Methods of Linear Algebra. Dover, New-York, 1959
- [13] P. Guillaume, *Nested multivariate Padé approximants*, Journ. of Comp. and Appl. Math., 82 (1997), 149-158.
- [14] P. Guillaume, *Convergence of the Nested multivariate Padé approximants*, Journal of Approximation Theory, 94, 455-566 (1998).
- [15] G. Heinig, K. Rost, *Algebraic Methods for Toeplitz-like matrices and operators*, Akademie-Verlag Berlin and Birkhauser Basel, Boston, Stuttgart 1984.

- [16] G. Heinig, P. Jankowski, K. Rost, *Fast Inversion Algorithms of Toeplitz-Plus-Hankel Matrices*, Numer. Math. 52, 665-682 (1988).
- [17] G. Heinig, V. Olshevski, *The Schur algorithm for matrices with Hessenberg displacement structure*, in V. Olshevski, editor, Structured Matrices in Mathematics, Computer Science and Engineering II, vol 281 of Contemporary Mathematics. AMS Publications (2001).
- [18] T.Kailath, *Fast reliable algorithms for matrices with structure*, in T.Kailath, A.H. Sayed, eds, SIAM Publications, Philadelphia 1999.
- [19] R. Hughes Jones and G.J. Makinson, *The generation of Chisholm rational polynomial approximants to power series in two variables*, J. Inst. Maths. Applics. (1974) 13, 299-310.
- [20] T. Kailath, V. Olshevski, *Displacement structure approach to discrete transform based preconditioners of G.Strang type and of T. Chan type*, Calcolo, 33 (1996), 191-208.
- [21] D. Levin, *General Order Padé type rational approximants defined from double power series*, J. Inst. Maths. Applics. (1976) 18, 1-8.
- [22] Ana C. Matos, *Recursive Computation of Padé-Legendre Approximants and Some Acceleration properties*, Numer.Math. 89, 535-560 (2001).
- [23] Ana C. Matos, Jeannette Van Iseghem, *Simultaneous Frobenius-Padé approximants*, Journ of Comp. and Appl. Math. 176 (2005) 231-258.