SIAM J. NUMER. ANAL.
Vol. 24, No. 1, February 1987

019

A RECURSIVE COMPUTATION SCHEME FOR MULTIVARIATE RATIONAL INTERPOLANTS*

A. CUYT†

Abstract. We derive here a recursive computation scheme for the rational interpolation method introduced in [7]. Explicit formulas for these multivariate rational interpolants are repeated in § 2 while the recursive algorithm is described in § 3. A number of interesting special cases such as the univariate rational interpolation problem and the multivariate Padé approximants introduced in [6] and [10] are dealt with in § 4. For some of these rational approximants other recursive schemes were described previously. Finally § 5 contains the numerical results: the multivariate rational interpolants described here are compared with multivariate polynomial interpolants, interpolating branched continued fractions introduced by Cuyt and Verdonk [8], interpolating branched continued fractions introduced by Siemaszko [12] and several multivariate Padé approximants [6], [3], [10]. Besides the fact that our multivariate rational interpolants allow a large degree of freedom in the choice for the numerator and denominator in order to fit the function to be approximated as well as possible, they also produce very accurate numerical results.

Key words. rational interpolation, multivariate, bivariate, Newton–Padé, E-algorithm

AMS(MOS) subject classifications. 41A05, 41A20, 41A63

1. Motivation. When we are dealing with interpolation problems, we must specify whether we are interested in an explicit formula for the interpolant or only in its value at some points different from the interpolation points. The former gives rise to a "coefficient problem," while the latter is a "value problem." For univariate rational interpolation the coefficient problem is translated into two linear systems of equations, one specifying the numerator coefficients and another which is homogeneous and determines the denominator coefficients. On the other hand, the values of the rational interpolant can be computed recursively by means of a generalization of the ε-algorithm which is a special case of the E-algorithm.

For multivariate rational interpolation the coefficient problem was solved in [7]; as in the univariate case, the unknown coefficients can be obtained from two linear systems of equations, one of which is homogeneous. We shall present here a recursive computation scheme for the calculation of the function values of these rational interpolants; the reasoning is again based on the E-algorithm.

2. Determinant formulas. Let us restrict everything to the case of two variables for the sake of simplicity. Furthermore we assume that none of the interpolation points in $\{(x_i, y_j)\}_{(i,j)\in\mathbb{N}^2}$ coincide and that the finite interpolation set $I=\{(i,j)\,|\,f_{ij}$ is given at $(x_i, y_j)\}$ is structured so that it satisfies the inclusion property. This means that if a point belongs to the data set, then the rectangular subset of points emanating from the origin with the given point as its furthermost corner also lies in the data set. In [7] it was illustrated that more general problems can be treated and that the formulas we give here in (3) remain valid. We could also deal with those more complicated situations here, but they only complicate the notation.

Let us first summarize the theory that solves the coefficient problem, because its solution will be the starting point for the construction of the recursive formulas.

* Received by the editors on April 26, 1985, and in final form January 19, 1986.

† Department of Mathematics and Computer Science, University of Antwerp, Universiteitsplein 1, B-2610 Antwerp, Belgium. This author was a research Fellow of the Alexander von Humboldt Foundation at the University of Bonn.

Consider the following set of basis functions for the real valued polynomials in two variables

$$B_{ij}(x, y) = \prod_{k=0}^{i-1} (x - x_k) \prod_{l=0}^{j-1} (y - y_l).$$

Clearly $B_{ij}(x, y)$ is a bivariate polynomial of degree $i+j$. Given the f_{ij}, we can write the interpolating series in a purely formal manner as

$$f(x, y) = \sum_{(i,j)\in\mathbb{N}^2} f_{0i,0j} B_{ij}(x, y)$$

where the $f_{0i,0j}$ are the bivariate divided differences

$$f_{0i,0j} = f[x_0, \cdots, x_i][y_0, \cdots, y_j]$$

which are given by

$$f[x_0, \cdots, x_i][y_0, \cdots, y_j] = \frac{f[x_1, \cdots, x_i][y_0, \cdots, y_j] - f[x_0, \cdots, x_{i-1}][y_0, \cdots, y_j]}{x_i - x_0},$$

or

$$f[x_0, \cdots, x_i][y_0, \cdots, y_j] = \frac{f[x_0, \cdots, x_i][y_1, \cdots, y_j] - f[x_0, \cdots, x_i][y_0, \cdots, y_{j-1}]}{y_j - y_0},$$

with

$$f[x_i][y_j] = f_{ij}.$$

In order to construct rational interpolants for the given set $I = \{(i, j) \mid f_{ij} \text{ is given at } (x_i, y_j)\}$, we choose two finite index sets N, a subset of I, and D, a subset of $\mathbb{N}^2$, and put as in [7]

$$p(x, y) = \sum_{(i,j)\in N} a_{ij} B_{ij}(x, y),$$

$$q(x, y) = \sum_{(i,j)\in D} b_{ij} B_{ij}(x, y),$$

(1) $$(f \cdot q - p)(x, y) = \sum_{(i,j)\in\mathbb{N}^2\setminus I} c_{ij} B_{ij}(x, y).$$

The rational interpolant $(p/q)(x, y)$ will then be denoted by

$$[N/D]_I.$$

Let us introduce a numbering $r(i, j)$ of the points in $\mathbb{N}^2$ based on the enumeration

$$(0,0), \underbrace{(1,0),(0,1)}_{\text{first diagonal}}, \underbrace{(2,0),(1,1),(0,2)}_{\text{second diagonal}}, \underbrace{(3,0),(2,1),(1,2),(0,3)}_{\text{third diagonal}}, \cdots$$

so that

$$r(i, j) = \frac{(i+j)(i+j+1)}{2} + j - i.$$

If we denote

$$\# N = n + 1,$$

then we can write

$$N = \bigcup_{l=0}^{n} N_l$$

with

$$\emptyset = N_{-1} \subset N_0 \subset N_1 \subset \cdots \subset N_{n-1} \subset N_n = N,$$
$$\# N_l = l+1,$$
$$N_l \backslash N_{l-1} = \{(i_l, j_l)\}, \qquad l = 0, 1, \cdots, n,$$
$$r(i_l, j_l) > r(i_r, j_r), \qquad l > r.$$

In other words, for each $l = 0, \cdots, n$ we add to N_{l-1} the point (i_l, j_l) which is the next in line in $N \cap \mathbb{N}^2$ according to the enumeration given above. Denote

$$\# D = m+1,$$

and proceed in the same way. Hence

$$D = \bigcup_{l=0}^{m} D_l,$$

with

$$D_{-1} = \emptyset, \quad D_l \backslash D_{l-1} = \{(d_l, e_l)\}, \quad l = 0, \cdots, m.$$

Since (1) can be rewritten as

$$\text{(2)} \qquad \begin{aligned} &(f \cdot q)_{0i,0j} = p_{0i,0j} = a_{ij}, \qquad (i,j) \in N, \\ &(f \cdot q)_{0i,0j} = 0, \qquad (i,j) \in I \backslash N, \end{aligned}$$

we will assume that the interpolation set I is such that exactly m of the homogeneous equations (2) are linearly independent. It is obvious that this condition guarantees the existence of a nontrivial solution of (2) given by the following determinant expressions, because the number of unknowns in the homogeneous system is now one more than its rank. We group the respective m elements in $I \backslash N$ that supply the linearly independent equations in the set H and number them also following the enumeration given above,

$$H = \bigcup_{l=1}^{m} H_l \subseteq I \backslash N,$$

with

$$H_0 = \emptyset, \quad H_l \backslash H_{l-1} = \{(h_l, k_l)\}, \qquad l = 1, \cdots, m.$$

The polynomials $p(x, y)$ and $q(x, y)$ satisfying (1) are then given by [7]

$$\text{(3a)} \qquad p(x,y) = \begin{vmatrix} \sum_{(i,j)\in N} f_{d_0 i, e_0 j} B_{ij}(x,y) & \cdots & \sum_{(i,j)\in N} f_{d_m i, e_m j} B_{ij}(x,y) \\ f_{d_0 h_1, e_0 k_1} & \cdots & f_{d_m h_1, e_m k_1} \\ \vdots & & \vdots \\ f_{d_0 h_m, e_0 k_m} & \cdots & f_{d_m h_m, e_m k_m} \end{vmatrix},$$

$$\text{(3b)} \qquad q(x,y) = \begin{vmatrix} B_{d_0 e_0}(x,y) & \cdots & B_{d_m e_m}(x,y) \\ f_{d_0 h_1, e_0 k_1} & \cdots & f_{d_m h_1, e_m k_1} \\ \vdots & & \vdots \\ f_{d_0 h_m, e_0 k_m} & \cdots & f_{d_m h_m, e_m k_m} \end{vmatrix}$$

where

$$f_{d_i h_j, e_i k_j} = f[x_{d_i}, \cdots, x_{h_j}][y_{e_i}, \cdots, y_{k_j}],$$

with

$$f_{d_i h_j, e_i k_j}=0 \quad \text{if } d_i>h_j \quad \text{or } e_i>k_j.$$

3. Recursive algorithm. The formulas (3) can be rewritten as follows. Multiply the $(l+1)$th row in $p(x,y)$ and $q(x,y)$ by $B_{h_l k_l}(x,y)$ $(l=1,\cdots,m)$, and then divide the $(l+1)$th column by $B_{d_l e_l}(x,y)$ $(l=0,\cdots,m)$. This results in

$$p(x,y)=\begin{vmatrix} \sum_{(i,j)\in N} f_{d_0 i, e_0 j} B_{d_0 i, e_0 j}(x,y) & \cdots & \sum_{(i,j)\in N} f_{d_m i, e_m j} B_{d_m i, e_m j}(x,y) \\ f_{d_0 h_1, e_0 k_1} B_{d_0 h_1, e_0 k_1}(x,y) & \cdots & f_{d_m h_1, e_m k_1} B_{d_m h_1, e_m k_1}(x,y) \\ \vdots & & \vdots \\ f_{d_0 h_m, e_0 k_m} B_{d_0 h_m, e_0 k_m}(x,y) & \cdots & f_{d_m h_m, e_m k_m} B_{d_m h_m, e_m k_m}(x,y) \end{vmatrix},$$

$$q(x,y)=\begin{vmatrix} 1 & \cdots & 1 \\ f_{d_0 h_1, e_0 k_1} B_{d_0 h_1, e_0 k_1}(x,y) & \cdots & f_{d_m h_1, e_m k_1} B_{d_m h_1, e_m k_1}(x,y) \\ \vdots & & \vdots \\ f_{d_0 h_m, e_0 k_m} B_{d_0 h_m, e_0 k_m}(x,y) & \cdots & f_{d_m h_m, e_m k_m} B_{d_m h_m, e_m k_m}(x,y) \end{vmatrix}$$

where for $k \leqq i$ and $l \leqq j$

$$B_{ki,lj}(x,y)=\frac{B_{ij}(x,y)}{B_{kl}(x,y)}=(x-x_k)\cdots(x-x_{i-1})(y-y_l)\cdots(y-y_{j-1}),$$

and for $k>i$ or $l>j$

$$f_{ki,lk}=0.$$

We can now easily construct $(m+1)$ series of which the successive partial sums can be found in the columns of $p(x,y)$. Take

$$t_0(0)=f_{d_0 i_0, e_0 j_0} B_{d_0 i_0, e_0 j_0}(x,y),$$

$$\Delta t_0(l-1)=t_0(l)-t_0(l-1)=f_{d_0 i_l, e_0 j_l} B_{d_0 i_l, e_0 j_l}(x,y), \qquad l=1,\cdots,n.$$

Then

$$t_0(n)=\sum_{(i,j)\in N} f_{d_0 i, e_0 j} B_{d_0 i, e_0 j}(x,y).$$

The next terms are given by

$$\Delta t_0(n+l-1)=t_0(n+l)-t_0(n+l-1)=f_{d_0 h_l, e_0 k_l} B_{d_0 h_l, e_0 k_l}(x,y), \qquad l=1,\cdots,m.$$

Note that $\Delta t_0(l-1)=0$ as long as $i_l<d_0$ or $j_l<e_0$.

In this way we obtain the first column of $p(x,y)$. We can proceed in the same way for the other columns. Define for $r=1,\cdots,m$

$$t_r(0)=f_{d_r i_0, e_r j_0} B_{d_r i_0, e_r j_0}(x,y),$$

$$\Delta t_r(l-1)=t_r(l)-t_r(l-1)=f_{d_r i_l, e_r j_l} B_{d_r i_l, e_r j_l}(x,y), \qquad l=1,\cdots,n,$$

$$\Delta t_r(n+l-1)=t_r(n+l)-t_r(n+l-1)=f_{d_r h_l, e_r k_l} B_{d_r h_l, e_r k_l}(x,y), \qquad l=1,\cdots,m.$$

Hence

$$t_r(n)=\sum_{(i,j)\in N} f_{d_r i, e_r j} B_{d_r i, e_r j}(x,y)$$

and the $(r+1)$th column of $p(x, y)$ is obtained. Again $\Delta t_r(l-1)=0$ for $i_l<d_r$ or $j_l<e_r$. Consequently

$$
\text{(4a)}\qquad p(x,y)=\begin{vmatrix} t_0(n) & \cdots & t_m(n) \\ \Delta t_0(n) & \cdots & \Delta t_m(n) \\ \vdots & & \vdots \\ \Delta t_0(n+m-1) & \cdots & \Delta t_m(n+m-1) \end{vmatrix},
$$

$$
\text{(4b)}\qquad q(x,y)=\begin{vmatrix} 1 & \cdots & 1 \\ \Delta t_0(n) & \cdots & \Delta t_m(n) \\ \vdots & & \vdots \\ \Delta t_0(n+m-1) & \cdots & \Delta t_m(n+m-1) \end{vmatrix}.
$$

This quotient of determinants can easily be computed using the E-algorithm [1]:

$$
E_0^{(l)}=t_0(l), \qquad l=0,\cdots,n+m,
$$

$$
g_{0,r}^{(l)}=t_r(l)-t_{r-1}(l), \qquad r=1,\cdots,m, \quad l=0,\cdots,n+m,
$$

$$
\text{(5a)}\qquad E_r^{(l)}=\frac{E_{r-1}^{(l)}g_{r-1,r}^{(l+1)}-E_{r-1}^{(l+1)}g_{r-1,r}^{(l)}}{g_{r-1,r}^{(l+1)}-g_{r-1,r}^{(l)}}, \qquad l=0,1,\cdots,n, \quad r=1,2,\cdots,m,
$$

$$
\text{(5b)}\qquad g_{r,s}^{(l)}=\frac{g_{r-1,s}^{(l)}g_{r-1,r}^{(l+1)}-g_{r-1,s}^{(l+1)}g_{r-1,r}^{(l)}}{g_{r-1,r}^{(l+1)}-g_{r-1,r}^{(l)}}, \qquad s=r+1,r+2,\cdots.
$$

The values $E_r^{(l)}$ and $g_{r,s}^{(l)}$ are stored as in Table 1 and Table 2.

Then

$$
[N/D]_I=E_m^{(n)}.
$$

TABLE 1

$$
\begin{array}{ccccc}
E_0^{(0)} & & & & \\
 & E_1^{(0)} & & & \\
E_0^{(1)} & & \ddots & & \\
 & E_1^{(1)} & & E_m^{(0)} & \\
E_0^{(2)} & & & \vdots & \ddots \\
 & \vdots & & & E_{n+m}^{(0)} \\
\vdots & & & E_m^{(n)} & \\
 & E_1^{(n+m-1)} & & & \\
E_0^{(n+m)} & & & &
\end{array}
$$

TABLE 2

$$
\left.\begin{array}{c}
g_{0,1}^{(0)} \\ g_{0,1}^{(1)} \\ g_{0,1}^{(2)} \\ \vdots \\ g_{0,1}^{(n+m)}
\end{array}\right|
\begin{array}{cc}
g_{0,2}^{(0)} & \\ & g_{1,2}^{(0)} \\ g_{0,2}^{(1)} & \\ & g_{1,2}^{(1)} \\ g_{0,2}^{(2)} & \\ & \vdots \\ \vdots & \\ & g_{1,2}^{(n+m-1)} \\ g_{0,2}^{(n+m)} &
\end{array}
\Bigg|\cdots\Bigg|
\begin{array}{cccc}
g_{0,r}^{(0)} & & & \\ & g_{1,r}^{(0)} & & \\ g_{0,r}^{(1)} & & \ddots & \\ & g_{1,r}^{(1)} & & g_{r-1,r}^{(0)} \\ g_{0,r}^{(2)} & & & \vdots \\ & \vdots & & g_{r-1,r}^{(n+m-r+1)} \\ \vdots & & \cdot^{\cdot^{\cdot}} & \\ & g_{1,r}^{(n+m-1)} & & \\ g_{0,r}^{(n+m)} & & &
\end{array}
\Bigg|\cdots\Bigg|
\begin{array}{cc}
g_{0,m}^{(0)} & \\ & \ddots \\ g_{0,m}^{(1)} & \\ & g_{m-1,m}^{(0)} \\ & \vdots \\ \vdots & g_{m-1,m}^{(n+1)} \\ & \cdot^{\cdot^{\cdot}} \\ g_{0,m}^{(n+m)} &
\end{array}
$$

Since the solution $q(x, y)$ of (2) is unique, the value $E_m^{(n)}$ itself does not depend upon the numbering of the points within the sets N, D and H. But this numbering affects the interpolation conditions satisfied by the intermediate E-values.

THEOREM. *For* $l=0,\cdots,n$ *and* $r=0,\cdots,m$

$$E_r^{(l)}=[N_l/D_r]_{N_l\cup\underbrace{\{(i_{l+1},j_{l+1}),\cdots,(i_n,j_n),(h_1,k_1),\cdots,(h_{r-n+l},k_{r-n+l})\}}_{r\text{ points}}}.$$

Proof. The proof is obvious since we know from [1] that

$$E_r^{(l)}=\frac{\begin{vmatrix} t_0(l) & \cdots & t_r(l)\\ \Delta t_0(l) & \cdots & \Delta t_r(l)\\ \vdots & & \vdots\\ \Delta t_0(l+r-1) & \cdots & \Delta t_r(l+r-1)\end{vmatrix}}{\begin{vmatrix} 1 & \cdots & 1\\ \Delta t_0(l) & \cdots & \Delta t_r(l)\\ \vdots & & \vdots\\ \Delta t_0(l+r-1) & \cdots & \Delta t_r(l+r-1)\end{vmatrix}},$$

and from [7] that

$$[N_l/D_r]_{N_l\cup\{(i_{l+1},j_{l+1}),\cdots,(i_n,j_n),(h_1,k_1),\cdots,(h_{r-n+l},k_{r-n+l})\}}$$

$$=\frac{\begin{vmatrix}\sum_{(i,j)\in N_l} f_{d_0i,e_0j}B_{ij}(x,y) & \cdots & \sum_{(i,j)\in N_l} f_{d_ri,e_rj}B_{ij}(x,y)\\ f_{d_0i_{l+1},e_0j_{l+1}} & \cdots & f_{d_ri_{l+1},e_rj_{l+1}}\\ \vdots & & \vdots\end{vmatrix}}{\begin{vmatrix} B_{d_0e_0}(x,y) & \cdots & B_{d_re_r}(x,y)\\ f_{d_0i_{l+1},e_0j_{l+1}} & \cdots & f_{d_ri_{l+1},e_rj_{l+1}}\\ \vdots & & \vdots\end{vmatrix}}.$$

If $n-l>r$ then the interpolation set does not contain points of H but only the points $\{(i_0,j_0),\cdots,(i_l,j_l),(i_{l+1},j_{l+1}),\cdots,(i_r,j_r)\}$. □

If N is enlarged with elements of H or if D is enlarged, then new points of $\mathbb{N}^2$ should be added to H. The first $(m+1)$ columns of the E-table remain unchanged and only subsequent columns or diagonals must be computed.

If N or D are completely changed, then it may be necessary to restart the algorithm.

If N and D contain the origin and satisfy the inclusion property themselves, then the structure of the g-table simplifies since

$$t_r(l)=0,\qquad l=0,\cdots,\frac{(d_r+e_r)(d_r+e_r+1)}{2}+e_r-1.$$

We can tell from Table 3 that we get a band structure instead of a triangular table.

TABLE 3

$$\begin{array}{ccccc}
0 & & & & \\
 & \cdots & & & \\
\vdots & & 0 & & \\
 & \cdot^{\cdot^{\cdot}} & & g_{r-1,r}^{(0)} & \\
0 & & & \vdots & \\
 & & \cdot^{\cdot^{\cdot}} & & \\
 & g_{1,r}^{(r-2)} & & g_{r-1,r}^{(n+m-r+1)} & \\
g_{0,r}^{(r-1)} & \vdots & \cdot^{\cdot^{\cdot}} & & \\
\vdots & g_{1,r}^{(n+m-1)} & & & \\
g_{0,r}^{(n+m)} & & & &
\end{array}$$

4. Special cases. This multivariate theory, in which a rational interpolant can be obtained either explicitly by means of the formulas (3) or by its values via the algorithm (5), includes a number of interesting special cases.

(a) Univariate rational interpolants of degree n in the numerator and m in the denominator can be obtained by choosing

$$D=\{(d,0)\,|\,0\leqq d\leqq m\},$$
$$N=\{(i,0)\,|\,0\leqq i\leqq n\},$$
$$H\subset I\backslash N=\{(h,0)\,|\,n+1\leqq h\leqq n+m+s \text{ with } s\geqq 0\}$$

where the integer s is the number of linearly dependent interpolation conditions in $I\backslash N$. The E-algorithm then simplifies to an ε-like algorithm. For more information we refer the reader to [2] and [7].

(b) Consequently univariate Padé approximants can also be computed by letting all the interpolation points coincide. In this case the E-algorithm reduces to the ε-algorithm.

(c) Multivariate general order Padé approximants, introduced in [10], can now also be computed recursively by letting the multivariate interpolation points coincide with the origin. The basis functions and divided differences become

$$B_{ki,lj}(x,y)=x^{i-k}y^{j-l},$$
$$f_{ki,lj}=\frac{\partial^{i-k+j-l}f}{\partial x^{i-k}\,\partial y^{j-l}}\Big|_{(0,0)}.$$

The fact that a recursive computation scheme now exists for this type of approximants may result in a number of new applications, such as convergence acceleration or their use for the solution of systems of simultaneous nonlinear equations.

(d) The multivariate Padé approximants of order (n, m) introduced in [6], which prove to satisfy a large number of the classical univariate properties and which can already be calculated recursively by means of the ε-algorithm if the $\Delta t_r(n+l-1)$ are homogeneous forms of degree $n+l-r$, can now be computed in a different way by choosing Δt_r as described in the previous section. To this end we take

$$D=\{(d,e)\,|\,nm\leqq d+e\leqq nm+m\},$$
$$N=\{(i,j)\,|\,nm\leqq i+j\leqq nm+n\},$$
$$H\subset\{(h,k)\,|\,nm\leqq h+k\leqq nm+n+m+s \text{ with } s\geqq 0\}$$

where the integer s is related to the block-size of this multivariate Padé table. For more details see [6]. Explicit determinant formulas for these index sets, involving near-Toeplitz matrices, are given in [4].

5. Numerical results. Suppose we have to solve the following numerical problem. A bivariate function $f(x, y)$ is only known by its function values in a number of distinct points (x_i, y_j) and we need an approximation for the value of f in some other points (u_i, v_j). This problem can be solved by calculating the function value of an interpolatory function (polynomial or rational) with or without solving the coefficient problem. The bivariate Beta function $B(x, y)$ will serve as a concrete example here. It is defined by

$$B(x,y)=\frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$$

where Γ is the Gamma function. Singularities occur at $x=-k$ and $y=-k$ ($k=0,1,2,\cdots$) and zeros at $y=-x-k$ ($k=0,1,2,\cdots$). By means of the recurrence formulas

$$\Gamma(x+1)=x\Gamma(x), \qquad \Gamma(y+1)=y\Gamma(y)$$

for the Gamma function, we can write

$$B(x,y)=\frac{1+(x-1)(y-1)f(x,y)}{xy}.$$

We shall now compute several types of approximants $R(x,y)$ for $f(x,y)$ and compare the exact value $B(u_i, v_j)$ with the expression

$$\frac{1+(u_i-1)(v_j-1)R(u_i,v_j)}{u_i v_j}.$$

We shall use the following interpolation methods:

(a) Polynomial interpolation

$$R(x,y)=\sum_{i=0}^{n_1}\sum_{j=0}^{n_2} f[x_0,\cdots,x_i][y_0,\cdots,y_j]B_{ij}(x,y)$$

satisfying

$$(f-R)(x,y)=\sum_{\{(i,j)\mid i>n_1 \text{ or } j>n_2\}} c_{ij}B_{ij}(x,y).$$

(b) Symmetric branched continued fractions of the form [8]

$$R(x,y)=\varphi[x_0][y_0]+\sum_{k=1}^{n_{0,x}}\frac{x-x_{k-1}\;|}{|\;\varphi[x_0,\cdots,x_k][y_0]}+\sum_{k=1}^{n_{0,y}}\frac{y-y_{k-1}\;|}{|\;\varphi[x_0][y_0,\cdots,y_k]}$$

$$+\sum_{l=1}^{n}\frac{(x-x_{l-1})(y-y_{l-1})\;|}{|\;\varphi[x_0,\cdots,x_l][y_0,\cdots,y_l]+\sum_{k=l+1}^{n_{lx}}\dfrac{x-x_{k-1}\;|}{|\;\varphi[x_0,\cdots,x_k][y_0,\cdots,y_l]}+\sum_{k=l+1}^{n_{ly}}\dfrac{y-y_{k-1}\;|}{|\;\varphi[x_0,\cdots,x_l][y_0,\cdots,y_k]}}$$

where

$$\varphi[x_0][y_0]=f(x_0,y_0),$$

$$\varphi[x_0,\cdots,x_k][y_0]=\frac{x_k-x_{k-1}}{\varphi[x_0,\cdots,x_{k-2},x_k][y_0]-\varphi[x_0,\cdots,x_{k-2},x_{k-1}][y_0]},$$

$$\varphi[x_0][y_0,\cdots,y_k]=\frac{y_k-y_{k-1}}{\varphi[x_0][y_0,\cdots,y_{k-2},y_k]-\varphi[x_0][y_0,\cdots,y_{k-2},y_{k-1}]},$$

$$\varphi[x_0,\cdots,x_l][y_0,\cdots,y_l]=(x_l-x_{l-1})(y_l-y_{l-1})/\Delta^{(l)}\varphi,$$

with

$$\Delta^{(l)}\varphi=\varphi[x_0,\cdots,x_{l-2},x_l][y_0,\cdots,y_{l-2},y_l]-\varphi[x_0,\cdots,x_{l-2},x_{l-1}][y_0,\cdots,y_{l-2},y_l]$$
$$-\varphi[x_0,\cdots,x_{l-2},x_l][y_0,\cdots,y_{l-2},y_{l-1}]$$
$$+\varphi[x_0,\cdots,x_{l-2},x_{l-1}][y_0,\cdots,y_{l-2},y_{l-1}]$$

and for $k>l$

$$\varphi[x_0,\cdots,x_k][y_0,\cdots,y_l]$$
$$=\frac{x_k-x_{k-1}}{\varphi[x_0,\cdots,x_{k-2},x_k][y_0,\cdots,y_l]-\varphi[x_0,\cdots,x_{k-2},x_{k-1}][y_0,\cdots,y_l]},$$

$$\varphi[x_0,\cdots,x_l][y_0,\cdots,y_k]$$
$$=\frac{y_k-y_{k-1}}{\varphi[x_0,\cdots,x_l][y_0,\cdots,y_{k-2},y_k]-\varphi[x_0,\cdots,x_l][y_0,\cdots,y_{k-2},y_{k-1}]},$$

satisfying

$$(f-R)(x,y)=\sum_{\substack{\mathbb{N}^2\setminus\{(i,j)\mid 0\leqq i\leqq n, 0\leqq j\leqq n_{iy}\}\\ \setminus\{(j,i)\mid 0\leqq i\leqq n, 0\leqq j\leqq n_{ix}\}}} c_{ij}B_{ij}(x,y).$$

(c) Branched continued fractions of the form [12]

$$R(x,y)=\psi[x_0][y_0]+\sum_{k=1}^{n_0}\left.\frac{y-y_{k-1}}{\psi[x_0][y_0,\cdots,y_k]}\right|$$
$$+\sum_{l=1}^{n}\left.\frac{x-x_{l-1}}{\psi[x_0,\cdots,x_l][y_0]+\sum_{k=1}^{n_l}\left.\frac{y-y_{k-1}}{\psi[x_0,\cdots,x_l][y_0,\cdots,y_k]}\right|}\right|$$

with

$$\psi[x_0][y_0]=f(x_0,y_0),$$

$$\psi[x_0][y_0,\cdots,y_k]=\frac{y_k-y_{k-1}}{\psi[x_0][y_0,\cdots,y_{k-2},y_k]-\psi[x_0][y_0,\cdots,y_{k-2},y_{k-1}]},$$

$$\psi[x_0,\cdots,x_l][y_0]=\frac{x_l-x_{l-1}}{\psi[x_0,\cdots,x_{l-2},x_l][y_0]-\psi[x_0,\cdots,x_{l-2},x_{l-1}][y_0]},$$

and for $l\geqq 1$

$$\psi[x_0,\cdots,x_l][y_0,\cdots,y_k]$$
$$=\frac{y_k-y_{k-1}}{\psi[x_0,\cdots,x_l][y_0,\cdots,y_{k-2},y_k]-\psi[x_0,\cdots,x_l][y_0,\cdots,y_{k-2},y_{k-1}]},$$

satisfying

$$(f-R)(x,y)=\sum_{\mathbb{N}^2\setminus\{(i,j)\mid 0\leqq i\leqq n, 0\leqq j\leqq n_i\}} c_{ij}B_{ij}(x,y).$$

(d) Multivariate Padé approximants calculated by means of the ε-algorithm [6]

$$R(x,y)=\varepsilon_{2m}^{(n-m)},$$

with

$$\varepsilon_{-1}^{(k)}=0,\qquad k=0,1,\cdots,$$

$$\varepsilon_0^{(k)}=\sum_{i+j=0}^{k}\frac{1}{i!\,j!}\left.\frac{\partial^{i+j}f}{\partial x^i\,\partial y^j}\right|_{(0,0)}x^iy^j,\qquad k=0,\cdots,n+m,$$

$$\varepsilon_{2l}^{(-l-1)}=0,\qquad l=0,1,\cdots,$$

$$\varepsilon_{l+1}^{(k)}=\varepsilon_{l-1}^{(k+1)}+\frac{1}{\varepsilon_l^{(k+1)}-\varepsilon_l^{(k)}},\quad l=0,1,\cdots,\quad k=-\left[\frac{l+2}{2}\right],-\left[\frac{l+2}{2}\right]+1,\cdots$$

satisfying the conditions described in (d) of the previous section.

(e) Chisholm's Padé approximants [3]

$$R(x, y) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{n} a_{ij} x^i y^j}{\sum_{i=0}^{n} \sum_{j=0}^{n} b_{ij} x^i y^j}$$

where a_{ij} and b_{ij} are computed so that in the Taylor series development

$$(f - R)(x, y) = \sum_{(i,j) \in \mathbb{N}^2} c_{ij} x^i y^j,$$

we have

$$c_{ij} = 0, \qquad (i, j) \in \{(i, j) \mid 0 \leqq i + j \leqq 2n\},$$

$$c_{2n+1-j,j} + c_{j,2n+1-j} = 0, \qquad j = 1, \cdots, 2n.$$

(f) Levin's general order Padé approximants [10]

$$R(x, y) = \frac{\sum_{(i,j) \in N} a_{ij} x^i y^j}{\sum_{(i,j) \in D} b_{ij} x^i y^j},$$

which were mentioned in (c) of the previous section and which satisfy

$$(f - R)(x, y) = \sum_{(i,j) \in \mathbb{N}^2 \setminus I} c_{ij} x^i y^j.$$

Our choice for the sets N, D and I is:

$$N = \{(i, j) \mid 0 \leqq i + j \leqq n_1\} \cup \left\{ \left(\left\lfloor \frac{n_1 + 1}{2} \right\rfloor, \left\lfloor \frac{n_1 + 1}{2} \right\rfloor \right) \right\},$$

$$D = \{(i, j) \mid 0 \leqq i + j \leqq n_2\},$$

$$I = \{(i, j) \mid 0 \leqq i \leqq n_3, 0 \leqq j \leqq n_3\}.$$

(g) General order rational interpolants

$$R(x, y) = \frac{\sum_{(i,j) \in N} a_{ij} B_{ij}(x, y)}{\sum_{(i,j) \in D} b_{ij} B_{ij}(x, y)},$$

as given by (3) here and with the next choice for the index sets N, D and I:

$$N = \{(i, j) \mid 0 \leqq i + j \leqq n_1\} \cup \left\{ \left(\left\lfloor \frac{n_1 + 1}{2} \right\rfloor, \left\lfloor \frac{n_1 + 1}{2} \right\rfloor \right) \right\},$$

$$D = \{(i, j) \mid 0 \leqq i + j \leqq n_2\},$$

$$I = \{(i, j) \mid 0 \leqq i \leqq n_3, 0 \leqq j \leqq n_3\}.$$

In order to use the same amount of data for each method, we are going to take

(a) $n_1 = 5$ and $n_2 = 5$,
(b) $n = 5$ and $n_{ix} = 5 = n_{iy}$ for $i = 0, \cdots, 5$,
(c) $n = 5$ and $n_i = 5$ for $i = 0, \cdots, 5$,
(d) $n = 4$ and $m = 3$,
(e) $n = 3$,
(f) $n_1 = 5$, $n_2 = 4$ and $n_3 = 5$,
(g) $n_1 = 5$, $n_2 = 4$ and $n_3 = 5$.

For (a), (b), (c) and (g) the interpolation points are chosen to be

$$x_0 = 0.90, \quad x_1 = -0.85, \quad x_2 = 0.47, \quad x_3 = -0.54, \quad x_4 = 0.18, \quad x_6 = -0.23,$$

$$y_0 = 0.70, \quad y_1 = -0.77, \quad y_2 = 0.60, \quad y_3 = -0.45, \quad y_4 = 0.21, \quad y_5 = -0.35,$$

which amounts to 36 data points (x_i, y_j). For (d), (e) and (f), respectively, 36, 34 and 36 Taylor coefficients are given in order to compute the approximant, namely

$$\left.\frac{\partial^{i+j} f}{\partial x^i \, \partial y^j}\right|_{(0,0)} \quad \text{with } (i, j) \in \{(i, j) \mid 0 \leqq i+j \leqq 7\} \text{ for (d)},$$

$$\left.\frac{\partial^{i+j} f}{\partial x^i \, \partial y^j}\right|_{(0,0)} \quad \text{with } (i, j) \in \{(i, j) \mid 0 \leqq i+j \leqq 6\}$$

$$\cup \{(1, 6), (2, 5), \cdots, (5, 2), (6, 1)\} \text{ for (e)},$$

$$\left.\frac{\partial^{i+j} f}{\partial x^i \, \partial y^j}\right|_{(0,0)} \quad \text{with } (i, j) \in \{(i, j) \mid 0 \leqq i \leqq 5, 0 \leqq j \leqq 5\} \text{ for (f)}.$$

We take $(u_i, v_j) \in \{(-0.75, -0.75),\ (-0.50, -0.50),\ (-0.25, -0.25),\ (0.25, 0.25),\ (0.50, 0.50),\ (0.75, 0.75)\}$.

The rational interpolants in (b) and (c) are computed using a backward evaluation algorithm while the rational interpolants from (g) are computed using the algorithm given in § 3 here. For the Padé approximants in (d) the well-known ε-algorithm is used while the Padé approximants from (e) and (f) are calculated using a similar technique [5] as the one described in § 3. Of course one can also compute the approximants in (e) and (f) by means of older techniques used by the Canterbury-group [9] and Levin themselves [11]. The numerical results can be found in Table 4 below. All the computations were performed in floating point double precision arithmetic on a VAX 11-780 with an input of 12 significant decimal digits.

TABLE 4

	(−0.75, −0.75)	(−0.50, −0.50)	(−0.25, −0.25)	(0.25, 0.25)	(0.50, 0.50)	(0.75, 0.75)
(a)	11.	0.06	−6.75	7.45	3.14151	1.69
(b)	9.95	−0.001	−6.7770	7.416291	3.14159276	1.694426
(c)	9.95	0.003	−6.775	7.416295	3.14159290	1.694426
(d)	8.8	−0.07	−6.786	7.416307	3.14159269	1.69442617
(e)	7.	−0.14	−6.787	7.416310	3.14159269	1.69442617
(f)	5.3	−0.46	−6.84	7.4164	3.1415938	1.69442617
(g)	9.91	0.0002	−6.7776	7.416310	3.14159292	1.694426
$B(x, y)$	9.88839829	0.	−6.77770467	7.41629871	3.14159265	1.694426166

For all types of approximants, except (c), the choice for $R(x, y)$ was such that it was a symmetric function. This was done because $B(x, y)$ is symmetric. We notice that unsymmetric approximants yield worse numerical results. The polynomial approximants lose a number of significant digits because of the singularities of the Beta function. The ε-algorithm (d) and the other Padé approximants (e) and (f) get all their information at the origin, far from the points (u_i, v_j). This is a disadvantage in comparison with the interpolation methods. As a conclusion we can say that the general order rational interpolants (g) for which a computational scheme was introduced here, behave quite well.

Acknowledgment. I hereby want to express my sincere thanks to Prof. Dr. H. Werner for the valuable discussions we had.

REFERENCES

[1] C. BREZINSKI, *A general extrapolation algorithm*, Numer. Math., to appear.

[2] G. CLAESSENS, *A useful identity for the rational Hermite interpolation table*, Numer. Math., 29 (1978), pp. 227-231.

[3] J. CHISHOLM, *Rational approximants defined from double power series*, Math. Comput., 27 (1973), pp. 841-848.

[4] A. CUYT, *A comparison of some multivariate Padé approximants*, SIAM J. Math. Anal., 14 (1983), pp. 195-202.

[5] ———, *Multivariate Padé approximants revisited*, BIT, to appear.

[6] ———, *Padé Approximants for Operators: Theory and Applications*, Lecture Notes in Mathematics 1065, Springer-Verlag, Berlin, 1984.

[7] A. CUYT AND B. VERDONK, *General order Newton-Padé approximants for multivariate functions*, Numer. Math., 43 (1984), pp. 293-307.

[8] ———, *Multivariate rational interpolation*, Computing, 34 (1985), pp. 41-61.

[9] P. GRAVES MORRIS, R. HUGHES JONES AND G. MAKINSON, *The calculation of some rational approximants in two variables*, J. Inst. Math. Appl., 13 (1974), pp. 311-320.

[10] D. LEVIN, *General order Padé type rational approximants defined from double power series*, J. Inst. Math. Appl., 18 (1976), pp. 1-8.

[11] ———, *On accelerating the convergence of infinite double series and integrals*, Math. Comput., 35 (1980), pp. 1331-1345.

[12] W. SIEMASZKO, *Thiele type branched continued fractions for two-variable functions*, J. Comput. Appl. Math., 9 (1983), pp. 137-153.