Computing 34, 41–61 (1985)

Computing

Multivariate Rational Interpolation

Annie A. M. Cuyt and **Brigitte M. Verdonk,** Wilrijk

Received February 13, 1984

Abstract — Zusammenfassung

Multivariate Rational Interpolation. Many papers have already been published on the subject of multivariate polynomial interpolation and also on the subject of multivariate Padé approximation. But the problem of multivariate rational interpolation has only very recently been considered; we refer among others to [8] and [3].

The computation of a univariate rational interpolant can be done in various equivalent ways: one can calculate the explicit solution of the system of interpolatory conditions, or start a recursive algorithm, or calculate the convergent of a continued fraction.

In this paper we will generalize each of those methods from the univariate to the multivariate case. Although the generalization is simple, the equivalence of the computational methods is completely lost in the multivariate case. This was to be expected since various authors have already remarked [2, 7] that there is no link between multivariate Padé approximants calculated by matching the Taylor series and those obtained as convergents of a continued fraction.

AMS Subject Classifications: 41A05, 41A20, 41A63.

Key words: Multivariate functions, rational interpolation, branched continued fractions, multivariate Padé approximants, recursive calculation of interpolants, multivariate inverse differences.

Multivariate rationale Interpolation. Das multivariate polynomiale Interpolationsproblem sowie die multivariate Padé-Approximation sind schon einige Jahre alt, aber das multivariate rationale Interpolationsproblem ist noch verhältnismäßig jung [3, 8].

Für univariate Funktionen gibt es verschiedene äquivalente Algorithmen zur Berechnung vom rationalen Interpolant: die Lösung eines Gleichungssystems, die rekursive Berechnung oder die Berechnung eines Kettenbruchs.

Diese Algorithmen werden hier verallgemeinert auf multivariate Funktionen. Wir bemerken, daß sie nun nicht mehr equivalent sind. Diese Beobachtung ist auch schon von anderen Mathematikern gemacht worden für das multivariate Padé-Approximationsproblem [2, 7], das man auch auf verschiedene Weisen lösen kann.

1. Algorithms for Univariate Rational Interpolation

Let the univariate function $f(x)$ be given in the non-coincident interpolation points $\{x_0, x_1, x_2, \ldots\}$. We consider the following problems:

$$\text{calculate } \frac{P_j^{n,n}(x)}{Q_j^{n,n}(x)} = \frac{\sum_{i=0}^{n} a_i x^i}{\sum_{i=0}^{n} b_i x^i} \tag{1}$$

$$\text{such that } (f \cdot Q_j^{n,n} - P_j^{n,n})(x_k) = 0 \quad \text{for } k = j, \ldots, j+2n$$

and

$$\text{calculate } \frac{P_j^{n+1,n}(x)}{Q_j^{n+1,n}(x)} = \frac{\sum_{i=0}^{n+1} a_i x^i}{\sum_{i=0}^{n} b_i x^i} \tag{2}$$

$$\text{such that } (f \cdot Q_j^{n+1,n} - P_j^{n+1,n})(x_k) = 0 \quad \text{for } k = j, \ldots, j+2n+1.$$

We shall say that the rational function "interpolates" the given function and by this we shall mean, also in the sequel of the text, that numerator and denominator of the rational function satisfy some linear conditions like (1) or (2).

This does not imply that the irreducible form of the rational function actually interpolates the given function at all the data, because, by constructing the irreducible form, a polynomial and hence some interpolation conditions may be cancelled in numerator and denominator of the rational interpolant.

The next theorem can be proved for the solutions of the problems (1) and (2). We denote $f(x_k)$ by f_k.

Theorem 1.1: *The statements* (*a*), (*b*), (*c*) *and* (*d*) *are equivalent*:

(*a*) $\frac{P_j^{n,n}}{Q_j^{n,n}}(x)$ *and* $\frac{P_j^{n+1,n}}{Q_j^{n+1,n}}(x)$ *respectively satisfy* (1) *and* (2)

(*b*) $\frac{P_j^{n,n}}{Q_j^{n,n}}(x) =$

$$= \frac{\begin{vmatrix} f_j & x-x_j & f_j(x-x_j) & (x-x_j)^2 & f_j(x-x_j)^2 & \ldots & (x-x_j)^n & f_j(x-x_j)^n \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ f_{j+2n} & x-x_{j+2n} & & & & & & \end{vmatrix}}{\begin{vmatrix} 1 & x-x_j & f_j(x-x_j) & \ldots & (x-x_j)^n & f_j(x-x_j)^n \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & x-x_{j+2n} & & & & \end{vmatrix}}$$

and

$\frac{P_j^{n+1,n}}{Q_j^{n+1,n}}(x) =$

$$= \frac{\begin{vmatrix} f_j & x-x_j & f_j(x-x_j) & \ldots & (x-x_j)^n & f_j(x-x_j)^n & (x-x_j)^{n+1} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ f_{j+2n+1} & x-x_{j+2n+1} & & & & & \end{vmatrix}}{\begin{vmatrix} 1 & x-x_j & f_j(x-x_j) & \ldots & (x-x_j)^{n+1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x-x_{j+2n+1} & & & \end{vmatrix}}$$

(c) $P_j^{n,n}(x)=(x-x_j)\,p_j^{n,n-1}\,P_{j+1}^{n,n-1}(x)-(x-x_{j+2n})\,p_{j+1}^{n,n-1}\,P_j^{n,n-1}(x)$

$Q_j^{n,n}(x)=(x-x_j)\,p_j^{n,n-1}\,Q_{j+1}^{n,n-1}(x)-(x-x_{j+2n})\,p_{j+1}^{n,n-1}\,Q_j^{n,n-1}(x)$

and

$P_j^{n+1,n}(x)=(x-x_j)\,q_j^{n,n}\,P_{j+1}^{n,n}(x)-(x-x_{j+2n+1})\,q_{j+1}^{n,n}\,P_j^{n,n}(x)$

$Q_j^{n+1,n}(x)=(x-x_j)\,q_j^{n,n}\,Q_{j+1}^{n,n}(x)-(x-x_{j+2n+1})\,q_{j+1}^{n,n}\,Q_j^{n,n}(x)$

where

$P_j^{n,n}=p_j^{n,n}\,x^n+\dots$

$Q_j^{n,n}=q_j^{n,n}\,x^n+\dots$

and

$P_j^{n+1,n}=p_j^{n+1,n}\,x^{n+1}+\dots$

$Q_j^{n+1,n}=q_j^{n+1,n}\,x^n+\dots$

(d) $\dfrac{P_j^{n,n}}{Q_j^{n,n}}(x)$ *and* $\dfrac{P_j^{n+1,n}}{Q_j^{n+1,n}}(x)$ *are respectively the* $(2n)$*-th and* $(2n+1)$*-th convergents of the continued fraction*

$$\varphi[x_j]+\sum_{k=0}^{\infty}\left|\frac{x-x_{j+k}}{\varphi[x_j,\dots,x_{j+k+1}]}\right.=$$

$$\varphi[x_j]+\cfrac{x-x_j}{\varphi[x_j,x_{j+1}]+\cfrac{x-x_{j+1}}{\varphi[x_j,x_{j+1},x_{j+2}]+\cfrac{x-x_{j+2}}{\dots}}}$$

where

$\varphi[x_j]=f(x_j)$

$$\varphi[x_j,\dots,x_{j+k+1}]=\frac{x_{j+k+1}-x_{j+k}}{\varphi[x_j,\dots,x_{j+k-1},x_{j+k+1}]-\varphi[x_j,\dots,x_{j+k-1},x_{j+k}]}.$$

A proof of this theorem can be constructed from [9], [12] and [5], of course under the assumption that none of the denominator determinants vanish. We also remark the important fact that the recursive algorithm described in [6] is merely a reformulation of (c) in order to obtain a generalization of the Aitken-Neville algorithm for polynomial interpolation. We shall now generalize the formulas (b), (c) and (d) for multivariate functions in the sections 2, 3 and 4 respectively. These generalizations shall be written down for the case of two variables, because the situation with more than two variables is only notationally more difficult. Numerical results can be found in section 5.

2. Multivariate Determinantal Formulas

Inspired by Wynn [12] who uses the determinantal formulas (b) of theorem 1.1 and by Thacher and Milne [10] who generalized a univariate polynomial interpolation scheme to the multivariate case, it is easy to write down the following rational interpolants for a function $f(x,y)$ given in the non-coincident points $\{(x_0, y_0), (x_1, y_1), (x_2, y_2) \ldots\}$. Let us denote $f(x_k, y_k)$ by f_k, $x - x_k$ by α_k and $y - y_k$ by β_k.

Theorem 2.1:

$$\frac{P_j^{n,n}}{Q_j^{n,n}}(x,y) =$$

$$= \frac{\begin{vmatrix} f_j & \alpha_j & \beta_j & f_j\alpha_j\; f_j\beta_j\; \alpha_j^2\; \alpha_j\beta_j\; \beta_j^2\; f_j\alpha_j^2\; f_j\alpha_j\beta_j\; f_j\beta_j^2 \ldots \alpha_j^n \ldots \beta_j^n\; f_j\alpha_j^n \ldots f_j\beta_j^n \\ \vdots & \vdots & \vdots & \\ f_{j+c(n,n)} & \alpha_{j+c(n,n)} & \beta_{j+c(n,n)} & \end{vmatrix}}{\begin{vmatrix} 1 & \alpha_j & \beta_j & f_j\alpha_j\; f_j\beta_j \ldots \quad \alpha_j^n \ldots \beta_j^n\; f_j\alpha_j^n \ldots f_j\beta_j^n \\ \vdots & \vdots & \vdots & \\ 1 & \alpha_{j+c(n,n)} & \beta_{j+c(n,n)} & \end{vmatrix}}$$

and

$$\frac{P_j^{n+1,n}}{Q_j^{n+1,n}}(x,y) =$$

$$= \frac{\begin{vmatrix} f_j & \alpha_j & \beta_j & f_j\alpha_j\; f_j\beta_j \ldots \alpha_j^n \ldots \beta_j^n\; f_j\alpha_j^n \ldots f_j\beta_j^n\; \alpha_j^{n+1} \ldots \beta_j^{n+1} \\ \vdots & \vdots & \vdots & \\ f_{j+c(n+1,n)} & \alpha_{j+c(n+1,n)} & \beta_{j+c(n+1,n)} & \end{vmatrix}}{\begin{vmatrix} 1 & \alpha_j & \beta_j & f_j\alpha_j\; f_j\beta_j \ldots \alpha_j^{n+1} \ldots \beta_j^{n+1} \\ \vdots & \vdots & \vdots & \\ 1 & \alpha_{j+c(n+1,n)} & \beta_{j+c(n+1,n)} & \end{vmatrix}}$$

are bivariate rational functions respectively of the form

$$\sum_{i+j=0}^{n} a_{ij} x^i y^j \Big/ \sum_{i+j=0}^{n} b_{ij} x^i y^j \quad \text{and} \quad \sum_{i+j=0}^{n+1} a_{ij} x^i y^j \Big/ \sum_{i+j=0}^{n} b_{ij} x^i y^j$$

satisfying respectively

$$(f \cdot Q_j^{n,n} - P_j^{n,n})(x_k, y_k) = 0 \quad \text{for} \quad k = j, \ldots, j + c(n,n)$$

and

$$(f \cdot Q_j^{n+1,n} - P_j^{n+1,n})(x_k, y_k) = 0 \quad \text{for} \quad k = j, \ldots, j + c(n+1,n)$$

where $c(n,n) = n^2 + 3n$ *and* $c(n+1,n) = n^2 + 4n + 2$.

Proof: We will prove the theorem for

$$\frac{P_j^{n,n}}{Q_j^{n,n}}(x,y)$$

because the reasoning for

$$\frac{P_j^{n+1,n}}{Q_j^{n+1,n}}(x,y)$$

is analogous.

Consider a rational function of the form

$$\frac{\sum_{i+j=0}^{n} a_{ij} x^i y^j}{\sum_{i+j=0}^{n} b_{ij} x^i y^j}.$$

It has n^2+3n+2 unknown coefficients a_{ij} and b_{ij}. So we need n^2+3n+1 conditions because one parameter can always be given by a normalization of numerator and denominator.

So consider the linear system of equations

$$(f \cdot Q_j^{n,n} - P_j^{n,n})(x_k, y_k) = 0 \text{ for } k=j, \ldots, j+c(n,n) \text{ with } c(n,n)=n^2+3n.$$

More explicitly

$$a_{00} - f_k b_{00} + a_{10} x_k + a_{01} y_k - f_k b_{10} x_k - f_k b_{01} y_k + \ldots + a_{n0} x_k^n + \ldots +$$
$$+ a_{0n} y_k^n - f_k b_{n0} x_k^n - \ldots - f_k b_{0n} y_k^n = 0$$

for $k=j, \ldots, j+c(n,n)$.

A solution of this homogeneous system of equations is given by:

$$Q_j^{n,n}(x,y) =$$

$$= \begin{vmatrix} 0 & 1 & 0 & 0 & x & y & 0 & 0 & 0 & x^2 & xy & y^2 & \ldots \\ 1 & f_j & x_j & y_j & f_j x_j & f_j y_j & x_j^2 & x_j y_j & y_j^2 & f_j x_j^2 & f_j x_j y_j & f_j y_j^2 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ 1 & f_{j+c(n,n)} & x_{j+c(n,n)} & y_{j+c(n,n)} & & & & & & & & & \end{vmatrix}$$

and

$$P_j^{n,n}(x,y) =$$

$$= \begin{vmatrix} 1 & 0 & x & y & 0 & 0 & x^2 & xy & y^2 & 0 & 0 & 0 & \ldots \\ 1 & f_j & x_j & y_j & f_j x_j & f_j y_j & x_j^2 & x_j y_j & y_j^2 & f_j x_j^2 & f_j x_j y_j & f_j y_j^2 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ 1 & f_{j+c(n,n)} & x_{j+c(n,n)} & y_{j+c(n,n)} & & & & & & & & & \end{vmatrix}$$

By making some suitable combinations of the rows and columns in the determinants above

$$\frac{P_j^{n,n}}{Q_j^{n,n}}(x,y)$$

can be written in the form given in theorem 2.1. ■

Remark the fact that, as in the univariate case, the degree of the denominator is not changed by going from

$$\frac{P_j^{n,n}}{Q_j^{n,n}}(x,y) \text{ to } \frac{P_j^{n+1,n}}{Q_j^{n+1,n}}(x,y)$$

and that the degree of the numerator is not altered by passing from

$$\frac{P_j^{n,n-1}}{Q_j^{n,n-1}}(x,y) \text{ to } \frac{P_j^{n,n}}{Q_j^{n,n}}(x,y).$$

When dealing with univariate rational interpolation, Wynn [12] suggested an algorithm for the recursive calculation of these determinants. It is based on the fact that only one row and column are added for the calculation of the next rational interpolant from the previous one. The use of this algorithm is doubtful here because in that way the number of terms in the recursion to calculate

$$\frac{P_j^{n,n}}{Q_j^{n,n}}(x,y) \text{ from } \frac{P_j^{n,n-1}}{Q_j^{n,n-1}}(x,y)$$

would be proportional to 2^{n+1}. So we suggest the recursive scheme described in section 3 which is a direct generalization of theorem 1.1 (c) but does however not calculate the determinants given in theorem 2.1.

3. Aitken-Neville-like Algorithm for Multivariate Rational Interpolants

The univariate algorithm described in theorem 1.1 (c) can be rewritten as follows:

$$\frac{P_j^{n+1,n}(x)}{Q_j^{n+1,n}(x)}=\frac{\begin{vmatrix} P_j^{n,n}(x) & (x-x_j)\,q_j^{n,n} \\ P_{j+1}^{n,n}(x) & (x-x_{j+2n+1})\,q_{j+1}^{n,n}\end{vmatrix}}{\begin{vmatrix} Q_j^{n,n}(x) & (x-x_j)\,q_j^{n,n} \\ Q_{j+1}^{n,n}(x) & (x-x_{j+2n+1})\,q_{j+1}^{n,n}\end{vmatrix}} \tag{4a}$$

and

$$\frac{P_j^{n,n}(x)}{Q_j^{n,n}(x)}=\frac{\begin{vmatrix} P_j^{n,n-1}(x) & (x-x_j)\,p_j^{n,n-1} \\ P_{j+1}^{n,n-1}(x) & (x-x_{j+2n})\,p_{j+1}^{n,n-1}\end{vmatrix}}{\begin{vmatrix} Q_j^{n,n-1}(x) & (x-x_j)\,p_j^{n,n-1} \\ Q_{j+1}^{n,n-1}(x) & (x-x_{j+2n})\,p_{j+1}^{n,n-1}\end{vmatrix}}. \tag{4b}$$

Consider for instance formula (4 a) and remark the following three important facts:

1. If we introduce the interpolation sets

$$S=\{x_j,\ldots,x_{j+2n+1}\}=S_1\cup S_2$$
$$S_1=\{x_j,\ldots,x_{j+2n}\}$$
$$S_2=\{x_{j+1},\ldots,x_{j+2n+1}\}$$

then

$$(f\,Q_j^{n+1,n}-P_j^{n+1,n})(x_k)=0 \text{ for } x_k \text{ in } S$$
$$(f\,Q_j^{n,n}-P_j^{n,n})(x_k)=0 \text{ for } x_k \text{ in } S_1$$
$$(f\,Q_{j+1}^{n,n}-P_{j+1}^{n,n})(x_k)=0 \text{ for } x_k \text{ in } S_2.$$

2. The set S_1 contains one and only one interpolationpoint which does not belong to S_2, namely x_j, and the same applies to S_2.
3. The coefficients $q_j^{n,n}$ and $q_{j+1}^{n,n}$ in (4 a) are chosen such that the degree of $Q_j^{n+1,n}(x)$ remains at most n.

Analogous conclusions can be written down for (4 b).

A first step to generalize (4 a) to the bivariate case, is to consider $n=0$.

It is easy to check that

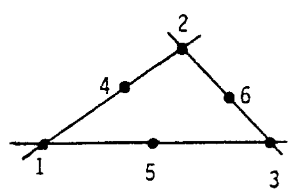
$$\frac{P_1^{1,0}}{Q_1^{1,0}}(x,y)=\frac{\begin{vmatrix} f_1 & x-x_1 & y-y_1 \\ f_2 & x-x_2 & y-y_2 \\ f_3 & x-x_3 & y-y_3 \end{vmatrix}}{\begin{vmatrix} 1 & x-x_1 & y-y_1 \\ 1 & x-x_2 & y-y_2 \\ 1 & x-x_3 & y-y_3 \end{vmatrix}}$$

is indeed a rational interpolant in the points (x_1, y_1), (x_2, y_2), (x_3, y_3), of degree 1 in the numerator and degree 0 in the denominator.

The coefficients of $x-x_j$, $y-y_j$ $(j=1,2,3)$ in the determinants are also the coefficients of the highest degree term in $Q_j^{0,0}(x,y)=1$ $(j=1,2,3)$ as is the case when we are dealing with a univariate function.

Now consider $n=1$ in (4 b). Suppose we are given the interpolationset

$$S=\{(x_1,y_1),(x_2,y_2),(x_3,y_3),(x_4,y_4),(x_5,y_5),(x_6,y_6)\}.$$



Then we can write $S=S_1 \cup S_2 \cup S_3$ with

$$S_1=\{(x_1,y_1),(x_4,y_4),(x_5,y_5)\}$$
$$S_2=\{(x_2,y_2),(x_4,y_4),(x_6,y_6)\}$$
$$S_3=\{(x_3,y_3),(x_5,y_5),(x_6,y_6)\}$$

where S_1 contains one and only one point which does not belong to S_2 or S_3, and so on.

Remark that the point (x_4, y_4) which belongs to S_1 and S_2 but not to S_3, lies on the straight line through (x_1, y_1) and (x_2, y_2). Similar results hold for (x_5, y_5) and (x_6, y_6). Since the points in S_1 are linearly independent and those in S_2 and S_3 too, we can calculate

$$\frac{P_{S_j}^{1,0}(x,y)}{Q_{S_j}^{1,0}(x,y)} \quad \text{for } j=1,2,3$$

where the subscript S_j now indicates that

$$(f\,Q_{S_j}^{1,0} - P_{S_j}^{1,0})(x_k, y_k) = 0 \text{ for } (x_k, y_k) \text{ in } S_j.$$

It is easy to see that a rational interpolant for the points in S can be given by

$$\frac{P_S^{(2)}}{Q_S^{(2)}}(x,y) = \frac{\begin{vmatrix} P_{S_1}^{1,0}(x,y) & p_1(x-x_1) & p_1(y-y_1) \\ P_{S_2}^{1,0}(x,y) & p_2(x-x_2) & p_2(y-y_2) \\ P_{S_3}^{1,0}(x,y) & p_3(x-x_3) & p_3(y-y_3) \end{vmatrix}}{\begin{vmatrix} Q_{S_1}^{1,0}(x,y) & p_1(x-x_1) & p_1(y-y_1) \\ Q_{S_2}^{1,0}(x,y) & p_2(x-x_2) & p_2(y-y_2) \\ Q_{S_3}^{1,0}(x,y) & p_3(x-x_3) & p_3(y-y_3) \end{vmatrix}}$$

where for instance p_j is the coefficient of x in $P_{S_j}^{1,0}(x,y)$ so that a minimal number of extra terms is added to $P_{S_j}^{1,0}(x,y)$ to obtain $P_S^{(2)}(x,y)$, $j=1,2,3$. The superscript (2) now indicates that, in our recursion, we have twice stepped down to rational functions of lower degree. In the same way $P_S^{0,0}/Q_S^{0,0}$ and $P_S^{1,0}/Q_S^{1,0}$ can respectively be denoted by $P_S^{(0)}/Q_S^{(0)}$ and $P_S^{(1)}/Q_S^{(1)}$.

For arbitrary n the recursion is formulated and proved in the next theorem.

Theorem 3.1: *Let the interpolation set* $S=\{(x_1,y_1),(x_2,y_2),\ldots,(x_{C^{(k)}},y_{C^{(k)}})\}$ *of* $C(k)$ *distinct points be given with* $C(k)\geq 2(k-1)+3$. *Suppose the set* S *of* $C(k)$ *interpolation points can be subdivided into* 3 *subsets* S_1, S_2, S_3 *satisfying the following additional conditions*:

(*a*) $S_1 \cup S_2 \cup S_3 = S$,

(*b*) $S_j\,(j=1,2,3)$ *contains one and only one point which does not belong to any other* S_i $(j\neq i)$; *by means of renumbering we can call this point* (x_j, y_j),

(*c*) *any point* (x_i, y_i) *does either belong to* $S_l\,(l=1,2,3)$ *or is a linear combination*

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \sum_{\substack{j=1 \\ j\neq l}}^{3} \alpha_j \begin{pmatrix} x_j \\ y_j \end{pmatrix} \text{ with } \sum_{\substack{j=1 \\ j\neq l}}^{3} \alpha_j = 1.$$

Then for $k=2n$:

$$P_S^{(2n)}(x,y) = \begin{vmatrix} P_{S_1}^{(2n-1)}(x,y) & p_1(x-x_1) & p_1(y-y_1) \\ P_{S_2}^{(2n-1)}(x,y) & p_2(x-x_2) & p_2(y-y_2) \\ P_{S_3}^{(2n-1)}(x,y) & p_3(x-x_3) & p_3(y-y_3) \end{vmatrix} \tag{5a}$$

and

$$Q_S^{(2n)}(x,y) = \begin{vmatrix} Q_{S_1}^{(2n-1)}(x,y) & p_1(x-x_1) & p_1(y-y_1) \\ Q_{S_2}^{(2n-1)}(x,y) & p_2(x-x_2) & p_2(y-y_2) \\ Q_{S_3}^{(2n-1)}(x,y) & p_3(x-x_3) & p_3(y-y_3) \end{vmatrix} \tag{5b}$$

are bivariate polynomials respectively of total degree $2n$ *and* $2n-1$ *satisfying* $(f\cdot Q_S^{(2n)} - P_S^{(2n)})(x_l, y_l) = 0$ *for* $l=1,\ldots,C(2n)$, *where* $p_j\,(j=1,2,3)$ *is a coefficient in* $P_{S_j}^{(2n-1)}(x,y)$ *chosen to minimize the number of terms in* $P_S^{(2n)}(x,y)$, *and for* $k=2n+1$

$$P_S^{(2n+1)}(x,y)=\begin{vmatrix} P_{S_1}^{(2n)}(x,y) & q_1(x-x_1) & q_1(y-y_1)\\ P_{S_2}^{(2n)}(x,y) & q_2(x-x_2) & q_2(y-y_2)\\ P_{S_3}^{(2n)}(x,y) & q_3(x-x_3) & q_3(y-y_3)\end{vmatrix}$$

and

$$Q_S^{(2n+1)}(x,y)=\begin{vmatrix} Q_{S_1}^{(2n)}(x,y) & q_1(x-x_1) & q_1(y-y_1)\\ Q_{S_2}^{(2n)}(x,y) & q_2(x-x_2) & q_2(y-y_2)\\ Q_{S_3}^{(2n)}(x,y) & q_3(x-x_3) & q_3(y-y_3)\end{vmatrix}$$

are bivariate polynomials respectively of total degree $2n+1$ *and* $2n$ *satisfying* $(f\cdot Q_S^{(2n+1)}-P_S^{(2n+1)})\,(x_l,y_l)=0$ *for* $l=1,\dots,C\,(2n+1)$ *where* q_j $(j=1,2,3)$ *is a coefficient in* $Q_{S_j}^{(2n)}(x,y)$ *chosen to minimize the number of terms in* $Q_S^{(2n+1)}(x,y)$.

Proof: We will only perform the proof for $k=2n$ because the case $k=2n+1$ is completely analogous. It is not difficult to see that in (x_1,y_1), (x_2,y_2) and (x_3,y_3) the interpolation condition is satisfied. In (x_l,y_l) with $3<l\le C\,(2n)$ the interpolation is proved as follows. The point (x_l,y_l) belongs to at least 2 of the S_j $(j=1,2,3)$ and we may renumber everything so that (x_l,y_l) belongs to S_1 and S_2. Suppose that (x_l,y_l) does not belong to S_3 (otherwise the proof is complete).

Since

$$(f\cdot Q_{S_1}^{(2n-1)})\,(x_l,y_l)=P_{S_1}^{(2n-1)}(x_l,y_l) \text{ and } (f\cdot Q_{S_2}^{(2n-1)})(x_l,y_l)=P_{S_2}^{(2n-1)}(x_l,y_l)$$

we may write

$$P_S^{(2n)}(x_l,y_l)=\begin{vmatrix} (f\,Q_{S_1}^{(2n-1)})(x_l,y_l) & p_1(x_l-x_1) & p_1(y_l-y_1)\\ (f\,Q_{S_2}^{(2n-1)})(x_l,y_l) & p_2(x_l-x_2) & p_2(y_l-y_2)\\ P_{S_3}^{(2n-1)}(x_l,y_l) & p_3(x_l-x_3) & p_3(y_l-y_3)\end{vmatrix}$$

$$=f(x_l,y_l)\left(Q_{S_1}^{(2n-1)}(x_l,y_l)\,p_2\,p_3\begin{vmatrix}(x_l-x_2) & (y_l-y_2)\\ (x_l-x_3) & (y_l-y_3)\end{vmatrix}-Q_{S_2}^{(2n-1)}(x_l,y_l)\,p_1\,p_3\begin{vmatrix}x_l-x_1 & y_l-y_1\\ x_l-x_3 & y_l-y_3\end{vmatrix}\right)$$

$$+P_{S_3}^{(2n-1)}(x_l,y_l)\,p_1\,p_2\begin{vmatrix}x_l-x_1 & y_l-y_1\\ x_l-x_2 & y_l-y_2\end{vmatrix}.$$

Since (x_l,y_l) does not belong to S_3, it lies on the straight line through (x_1,y_1) and (x_2,y_2) and so

$$\begin{vmatrix}x_l-x_1 & y_l-y_1\\ x_l-x_2 & y_l-y_2\end{vmatrix}=0.$$

Consequently

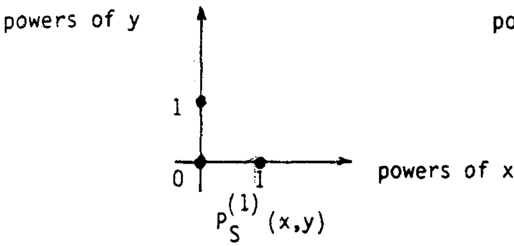
$$P_S^{(2n)}(x_l,y_l)=f(x_l,y_l)\cdot Q_S^{(2n)}(x_l,y_l).$$

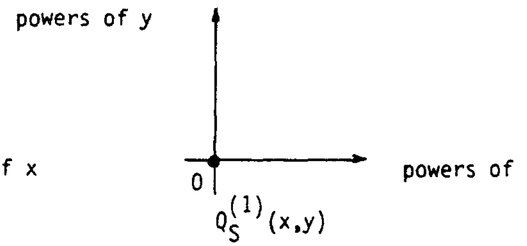
To prove the degrees of $P_S^{(2n)}(x,y)$, $Q_S^{(2n)}(x,y)$, $P_S^{(2n+1)}(x,y)$ and $Q_S^{(2n+1)}(x,y)$, we use induction.

We know that the degree of $P_S^{(1)}(x,y)$ is 1 and that of $Q_S^{(1)}(x,y)$ is 0. It is easy to see from (5 a) and (5 b) if we expand the determinants by the first column, that in each recursive step the degree of the numerator and the denominator is raised by one. ■

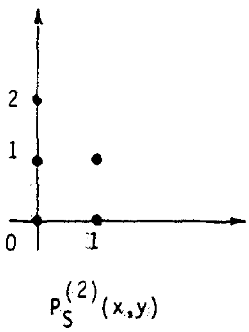
The lower bound for $C(2n)$ and $C(2n+1)$ in theorem 3.1 is a consequence of condition (b) in the formulation of the theorem: for $k>1$ the set $S\backslash S_j$ $(j=1,2,3)$ contains at least two points not belonging to S_j and for $k=1$ three interpolation points are needed to start the recursion. Remark also that the algorithm constructs a rather high degree rational function in order to fit rather few points.

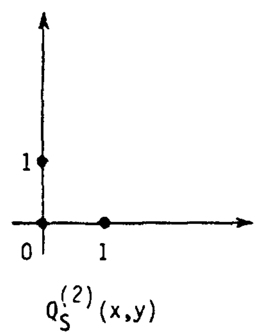
Let us now discuss the choice of the parameters p_j and q_j. We know that $C(1)=3$ and that in this case $P_S^{(1)}(x,y)$ and $Q_S^{(1)}(x,y)$ are bivariate polynomials of total degree respectively 1 and 0, which we shall indicate by means of the following figures:



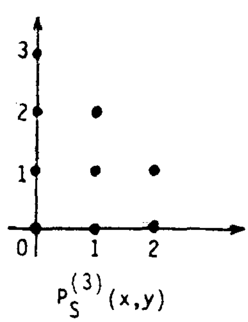


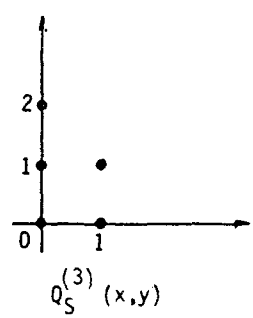
For S containing $C(2)$ interpolationpoints and in order to calculate $P_S^{(2)}(x,y)$ and $Q_S^{(2)}(x,y)$ with a minimal number of extra terms in $P_S^{(2)}(x,y)$ in comparison with $P_S^{(1)}(x,y)$, we have to choose p_j equal to the coefficient of x in $P_{S_j}^{(1)}(x,y)$. Then $P_S^{(2)}(x,y)$ and $Q_S^{(2)}(x,y)$ have the following form:





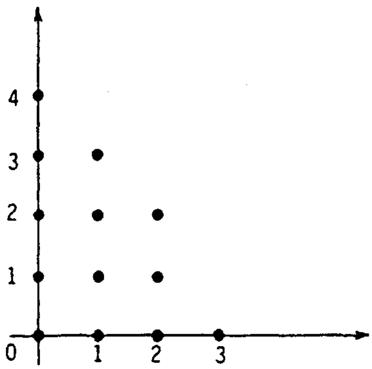
To calculate $P_S^{(3)}(x,y)$ and $Q_S^{(3)}(x,y)$ for a given S with $C(3)$ interpolationpoints so that we have a minimal number of terms in $Q_S^{(3)}(x,y)$, we now have to choose q_j equal to the coefficient of x in $Q_S^{(2)}(x,y)$. So $P_S^{(3)}(x,y)$ and $Q_S^{(3)}(x,y)$ have the following form:



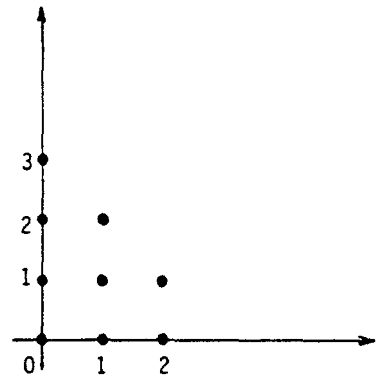


We repeat the same procedure for the calculation of the rational functions interpolating in S containing $C(k)$ interpolationpoints $(k>3)$. The $P_S^{(2n)}(x,y)/Q_S^{(2n)}(x,y)$ and $P_S^{(2n+1)}(x,y)/Q_S^{(2n+1)}(x,y)$ are then of the following form:

$k=4$: p_j=coefficient of x^2y in $P_{S_j}^{(3)}(x,y)$

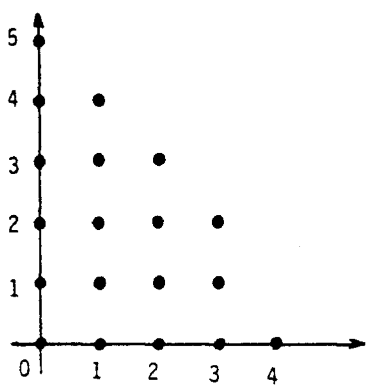


$P_S^{(4)}(x,y)$

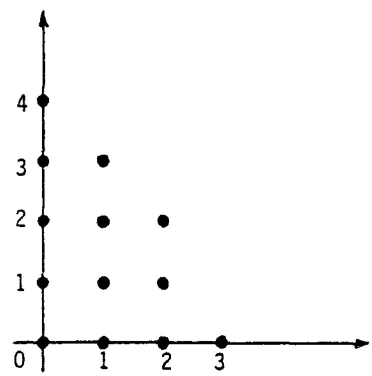


$Q_S^{(4)}(x,y)$

$k=5$: q_j=coefficient of x^2y in $Q_{S_j}^{(4)}(x,y)$



$P_S^{(5)}(x,y)$



$Q_S^{(5)}(x,y)$

$k=6$: p_j=coefficient of x^3y^2 in $P_{S_j}^{(5)}(x,y)$

and so on.

It is obvious, that, instead of eliminating high powers of x, we could also have eliminated high powers of y; we could as well have used a mixed strategy. What's more, the obtained rational interpolant does also depend on the partitioning of S into S_1, S_2, S_3.

For numerical results illustrating this method we refer to section 5. However, we want to remark here already that if the set S contains many points, then for the computation of $P_S^{(2n)}(x,y)/Q_S^{(2n)}(x,y)$ or $P_S^{(2n+1)}(x,y)/Q_S^{(2n+1)}(x,y)$ a lot of interpolating rational functions of lower degree are needed, which may cause some programming difficulties. Also one should preferably not choose the points (x_1, y_1), (x_2, y_2) and (x_3, y_3) – in the formulation of theorem 3.1 – on one straight line in order to avoid degeneracy of the determinants for the calculation of the next interpolating rational function.

4. Multivariate Branched Continued Fractions

We shall only consider the construction of continued fractions symmetric in the variables x and y because there is no reasonable argument for not doing so. The algorithm which we shall develop can, just as in the previous section, easily be used both for the calculation of the coefficients of the branched continued fraction and the computation of the function value of some convergent.

Given two sequences $\{x_0, x_1, x_2, \dots\}$ and $\{y_0, y_1, y_2, \dots\}$ of distinct real points, the branched continued fraction will interpolate $f(x,y)$ at the points in $\{x_0, x_1, x_2, \dots\} \times \{y_0, y_1, y_2, \dots\}$.

Theorem 4.1:

$$f(x,y) = \varphi[x_0][y_0] + \sum_{k=1}^{\infty} \frac{x - x_{k-1} \,|}{|\, \varphi[x_0,\dots,x_k][y_0]} + \sum_{k=1}^{\infty} \frac{y - y_{k-1} \,|}{|\, \varphi[x_0][y_0,\dots,y_k]}$$

$$+ \sum_{j=1}^{\infty} \frac{(x - x_{j-1})(y - y_{j-1}) \,|}{|\, \varphi[x_0,\dots,x_j][y_0,\dots,y_j] + \sum_{k=j+1}^{\infty} \frac{x - x_{k-1} \,|}{|\, \varphi[x_0,\dots,x_k][y_0,\dots,y_j]} + \sum_{k=j+1}^{\infty} \frac{y - y_{k-1} \,|}{|\, \varphi[x_0,\dots,x_j][y_0,\dots,y_k]}}$$

where the inverse differences are given by

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

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

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

$$\varphi[x_0,\dots,x_j][y_0,\dots,y_j] =$$

$$= \frac{(x_j - x_{j-1})(y_j - y_{j-1})}{\varphi[x_0,\dots,x_{j-2},x_j][y_0,\dots,y_{j-2},y_j] - \varphi[x_0,\dots,x_{j-2},x_{j-1}][y_0,\dots,y_{j-2},y_j] - \varphi[x_0,\dots,x_{j-2},x_j][y_0,\dots,y_{j-2},y_{j-1}] + \varphi[x_0,\dots,x_{j-2},x_{j-1}][y_0,\dots,y_{j-2},y_{j-1}]}$$

and for $k > j$

$$\varphi[x_0,\dots,x_k][y_0,\dots,y_j] =$$

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

$$\varphi[x_0,\dots,x_j][y_0,\dots,y_k] =$$

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

Proof: We assume that the reader is familiar with one-dimensional interpolatory continued fractions. So we can write

$$f(x,y)=\varphi[x][y]=\varphi[x_0][y]+\frac{x-x_0}{\varphi[x_0,x][y]}$$

$$=\varphi[x_0][y_0]+\frac{y-y_0}{\varphi[x_0][y_0,y]}+\frac{x-x_0}{\varphi[x_0,x][y]}$$

$$=\varphi[x_0][y_0]+\sum_{k=1}^{\infty}\left.\frac{y-y_{k-1}}{\varphi[x_0][y_0,\ldots,y_k]}\right|+\frac{x-x_0}{\varphi[x_0,x][y]}.$$

Let us introduce the function $g_0(x,y)$ by

$$(x-x_0)g_0(x,y)=\frac{x-x_0}{\varphi[x_0,x][y]}$$

where

$$g_0(x,y)=\frac{1}{\varphi[x_0,x][y]}.$$

By calculating inverse differences ξ_0 for g_0 we obtain

$$g_0(x,y)=\xi_0[x][y]=\xi_0[x][y_0]+\frac{y-y_0}{\xi_0[x][y_0,y]}=$$

$$=\frac{1}{\varphi[x_0,x_1][y_0]+\sum_{k=2}^{\infty}\left.\frac{x-x_{k-1}}{\varphi[x_0,\ldots,x_k][y_0]}\right|}+\frac{y-y_0}{h_0(x,y)}$$

where $h_0(x,y)=\xi_0[x][y_0,y]$.

So already

$$f(x,y)=\varphi[x_0][y_0]+\sum_{k=1}^{\infty}\left.\frac{y-y_{k-1}}{\varphi[x_0][y_0,\ldots,y_k]}\right|+$$

$$+\sum_{k=1}^{\infty}\left.\frac{x-x_{k-1}}{\varphi[x_0,\ldots,x_k][y_0]}\right|+\frac{(x-x_0)(y-y_0)}{h_0(x,y)}.$$

By computing inverse differences π_0 for h_0 we get

$$h_0(x,y)=\pi_0[x][y]=\pi_0[x_1][y]+\frac{x-x_1}{\pi_0[x_1,x][y]}$$

$$=\pi_0[x_1][y_1]+\frac{y-y_1}{\pi_0[x_1][y_1,y]}+\frac{x-x_1}{\pi_0[x_1,x][y]}$$

$$=\pi_0[x_1][y_1]+\sum_{k=2}^{\infty}\left.\frac{y-y_{k-1}}{\pi_0[x_1][y_1,\ldots,y_k]}\right|+\frac{x-x_1}{\pi_0[x_1,x][y]}$$

where

$$\begin{aligned}\pi_0[x][y]&=\xi_0[x][y_0,y]\\&=\frac{(y-y_0)(x-x_0)}{f(x,y)-f(x_0,y)-f(x,y_0)+f(x_0,y_0)}\\&=\varphi[x_0,x][y_0,y]\end{aligned}$$

and by induction

$$\pi_0[x_1][y_1,\dots,y_k]=\varphi[x_0,x_1][y_0,\dots,y_k]$$

$$\pi_0[x_1,x][y]=\varphi[x_0,x_1,x][y_0,y].$$

So we can write

$$\begin{aligned}h_0(x,y)&=\varphi[x_0,x_1][y_0,y_1]+\sum_{k=2}^{\infty}\left.\frac{y-y_{k-1}}{|\varphi[x_0,x_1][y_0,\dots,y_k]}\right|+\frac{(x-x_1)}{\varphi[x_0,x_1,x][y_0,y]}\\&=\varphi[x_0,x_1][y_0,y_1]+\sum_{k=2}^{\infty}\left.\frac{y-y_{k-1}}{|\varphi[x_0,x_1][y_0,\dots,y_k]}\right|+(x-x_1)g_1(x,y)\end{aligned}$$

where

$$g_1(x,y)=\frac{1}{\varphi[x_0,x_1,x][y_0,y]}.$$

If we introduce inverse differences ξ_1 for g_1 we can repeat the whole reasoning which provides us with a function h_1 and inverse differences π_1:

$$g_1(x,y)=\xi_1[x][y]=\xi_1[x][y_1]+\frac{y-y_1}{\xi_1[x][y_1,y]}$$

$$h_1(x,y)=\xi_1[x][y_1,y]=\pi_1[x_2][y_2]+\frac{y-y_2}{\pi_1[x_2][y_2,y]}+$$

$$+\frac{x-x_2}{\pi_1[x_2,x][y]}=\varphi[x_0,x_1,x_2][y_0,y_1,y_2]+\dots.$$

In this way we obtain the desired interpolatory continued fraction. ■

To obtain rational interpolants we are going to consider convergents of the branched continued fraction given in theorem 4.1. To indicate which convergent we compute we need a multi-index $\bar{n}=(n,i_{0x},i_{0y},\dots,i_{nx},i_{ny})$:

$$\begin{aligned}\frac{P_0^{\bar{n}}}{Q_0^{\bar{n}}}(x,y)&=\varphi[x_0][y_0]+\sum_{k=1}^{i_{0x}}\left.\frac{x-x_{k-1}}{|\varphi[x_0,\dots,x_k][y_0]}\right|+\sum_{k=1}^{i_{0y}}\left.\frac{y-y_{k-1}}{|\varphi[x_0][y_0,\dots,y_k]}\right|\\&+\sum_{j=1}^{n}\left.\frac{(x-x_{j-1})(y-y_{j-1})}{|\varphi[x_0,\dots,x_j][y_0,\dots,y_j]+\sum\limits_{k=j+1}^{i_{jx}}\left.\frac{x-x_{k-1}}{|\varphi[x_0,\dots,x_k][y_0,\dots,y_j]}\right|+\sum\limits_{k=j+1}^{i_{jy}}\left.\frac{y-y_{k-1}}{|\varphi[x_0,\dots,x_j][y_0,\dots,y_k]}\right|}\right|.\end{aligned}$$

For these rational functions the following interpolation property can be proved.

Theorem 4.2: *If the multi-index* $\bar{n}$ *is such that* $i_{0y} \geq i_{1y} \geq \ldots \geq i_{ny}$ *and* $i_{0x} \geq i_{1x} \geq \ldots \geq i_{nx}$ *then* $(f \cdot Q_0^{\bar{n}} - P_0^{\bar{n}})(x_{l_1}, y_{l_2}) = 0$ *for* (l_1, l_2) *belonging to*

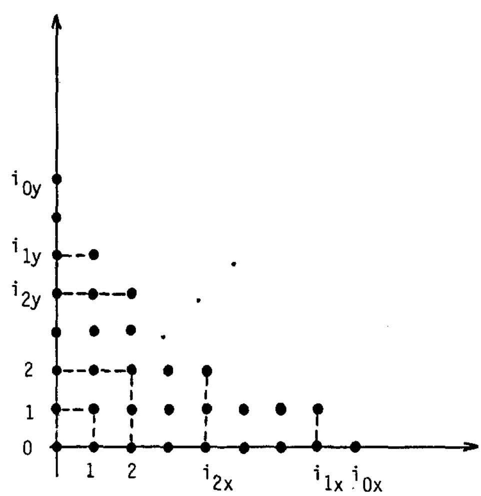
$$I = \bigcup_{j=0}^{n} \left(\{(j,k) \mid j \leq k \leq i_{jy}\} \cup \{(k,j) \mid j \leq k \leq i_{jx}\} \right).$$

Proof: It is easy to construct a proof from one-dimensional results if we group the interpolation points as follows:

$$(x_0, y_0), \underbrace{(x_1, y_0), \ldots, (x_{i_{0x}}, y_0)}_{i_0 x \text{ points}}, \underbrace{(x_0, y_1), \ldots, (x_0, y_{i_{0y}})}_{i_{0y} \text{ points}},$$
$$(x_1, y_1), \underbrace{(x_2, y_1), \ldots, (x_{i_{1x}}, y_1)}_{i_{1x}-1 \text{ points}}, \underbrace{(x_1, y_2), \ldots, (x_1, y_{i_{1y}})}_{i_{1y}-1 \text{ points}}, \ldots,$$
$$(x_n, y_n), \underbrace{(x_{n+1}, y_n), \ldots, (x_{i_{nx}}, y_n)}_{i_{nx}-n \text{ points}}, \underbrace{(x_n, y_{n+1}), \ldots, (x_n, y_{i_{ny}})}_{i_{ny}-n \text{ points}}$$

and take into account the form of the continued fraction in theorem 4.1.

To clarify the numbering we make a drawing of the set I in $\mathbb{N}^2$:



■

The subscript zero in $P_0^{\bar{n}}$ and $Q_0^{\bar{n}}$ refers to the fact that we are constructing rational functions interpolating a set of points starting with (x_0, y_0). The whole construction can also be used for the calculation of rational interpolants starting at (x_j, y_j), which are then called $\frac{P_j^{\bar{n}}}{Q_j^{\bar{n}}}(x, y)$.

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).

When using interpolation this problem can be attacked in two ways:

(A) either by calculating the coefficients of the interpolatory function (polynomial, rational, ...) and evaluating this function in the points (u_i, v_j)

(B) or by calculating the function value of the interpolatory function at (u_i, v_j) without knowledge of all its coefficients.

We shall compare the following methods of class (A):

– Newton-interpolation [3, 11]:

$$p(x,y)=\sum_{i+j=0}^{n} f[x_0,\ldots,x_i][y_0,\ldots,y_j]\,B_{ij}(x,y)$$

with

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

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

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

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

where $B_{00}(x,y)=1$.

– branched continued fractions (on a large number of data) introduced here in section 4.

– Padé-approximants introduced by Chisholm [1] of the type

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

where

$$(f\cdot q-p)(x,y)=\sum_{(i,j)\in\mathbb{N}^2\setminus E} d_{ij}x^i y^j$$

for

$$E=\{(i,j)\mid 0\le i+j\le 2n\}$$

and where $d_{2n+1-l,l}+d_{l,2n+1-l}=0$ for $l=1,\ldots,n$; here $f(x,y)$ is given by part of its Taylor series expansion

$$\sum_{i+j=0}^{\infty} c_{ij} x^i y^j$$

instead of by its function values in a number of points.

We shall also compare the following methods of class (B):

– branched continued fractions (on several small datasets) introduced here in section 4.

– rational interpolants constructed as in section 3; the coefficients p_1, p_2, p_3 and q_1, q_2, q_3 can also be calculated recursively without the knowledge of all the other coefficients.

– Padé-approximants calculated by means of the ε-algorithm [2]:

For

$$\varepsilon_{-1}^{(k)} = 0, \quad k = 0, 1, \ldots$$
$$\varepsilon_0^{(k)} = \sum_{i+j=0}^{k} c_{ij} x^i y^j$$

where $f(x, y)$ is given by part of its Taylor series expansion

$$\sum_{i+j=0}^{\infty} c_{ij} x^i y^j$$
$$\varepsilon_{2l}^{(-l-1)} = 0 \qquad l = 0, 1, \ldots$$
$$\varepsilon_{l+1}^{(k)} = \varepsilon_{l-1}^{(k+1)} + \frac{1}{\varepsilon_l^{(k+1)} - \varepsilon_l^{(k)}} \qquad \begin{array}{l} l = 0, 1, \ldots \\ k = -l, -l+1, \ldots \end{array}$$

we have

$$\varepsilon_{2n}^{(0)} = \frac{p(x, y)}{q(x, y)} = \frac{\sum_{i+j=n^2}^{n^2+n} a_{ij} x^i y^j}{\sum_{i+j=n^2}^{n^2+n} b_{ij} x^i y^j}$$

satisfying

$$(f \cdot q - p)(x \cdot y) = \sum_{i+j=n^2+2n+1}^{\infty} d_{ij} x^i y^j.$$

The methods of class (A) could be referred to as "global" because they will interpolate the function $f(x, y)$ simultaneously at a large number of data spread over the whole region; therefore we calculate and store the coefficients of the interpolatory function and evaluate that same function several times, i.e. at each point (u_i, v_j).

The methods of class (B) on the contrary could be referred to as "local", the branched continued fractions and the rational interpolants calculated recursively by means of determinants will each interpolate $f(x, y)$ at a smaller number of data in the neighbourhood of every point (u_i, v_j) while the ε-algorithm does use the same Taylor-coefficients at the different points (u_i, v_j) but has indeed to be restarted when going from one point to another.

The bivariate Beta function $B(x, y)$ will serve as a concrete example to illustrate all these approximation methods. It is defined by

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

where Γ is the Gamma function. Singularities occur for $x=-k$ and $y=-k$ $(k=0,1,2,\ldots)$ and zeros for $y=-x-k$ $(k=0,1,2,\ldots)$.

By means of the recurrence formulas

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

for the Gamma function, we can write

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

where it is possible to calculate a Taylor series expansion for $f(x-1, y-1)$ by the first method suggested in [4]; this Taylor series expansion is necessary to compute Chisholm's diagonal Padé approximants and also to start the ε-algorithm. The other interpolation schemes use function evaluations.

All the considered types of approximants $R(x, y)$, polynomial as well as rational and global as well as local shall be computed for the function $f(x-1, y-1)$ and afterwards we shall compare the exact value of $B(x, y)$ with the expression

$$\frac{1+(x-1)(y-1)\,R(x, y)}{x\,y}.$$

In order to use approximately the same amount of data for each method of class (A), we are going to consider

$$-\sum_{i=0}^{5}\sum_{j=0}^{5} f[x_0, \ldots, x_i][y_0, \ldots, y_j]\, B_{ij}(x, y) \quad \text{(36 data)}$$

$$-\varphi[x_0][y_0]+\sum_{k=1}^{5}\left.\frac{x-x_{k-1}}{\varphi[x_0, \ldots, x_k][y_0]}\right|+\sum_{k=1}^{5}\left.\frac{y-y_{k-1}}{\varphi[x_0][y_0, \ldots, y_k]}\right|$$

$$+\sum_{j=1}^{5}\left.\frac{(x-x_{j-1})(y-y_{j-1})}{\varphi[x_0, \ldots, x_j][y_0, \ldots, y_j]+\sum_{k=j+1}^{5}\left.\frac{x-x_{k-1}}{\varphi[x_0, \ldots, x_k][y_0, \ldots, y_j]}\right|+\sum_{k=j+1}^{5}\left.\frac{y-y_{k-1}}{\varphi[x_0, \ldots, x_j][y_0, \ldots, y_k]}\right|}\right|$$

(36 data).

– Chisholm's Padé approximants with $n=3$ (34 data).

For (u_i, v_j) equal to $(-0.75, -0.75)$, $(-0.50, -0.50)$, $(-0.25, -0.25)$, $(0.25, 0.25)$, $(0.50, 0.50)$ or $(0.75, 0.75)$ we shall take $\{x_0, x_1, \ldots, x_5\}=\{0.76, 0.48, 0.23, -0.22, -0.48, -0.73\}$ and $\{y_0, y_1, \ldots, y_5\}=\{0.74, 0.49, 0.22, -0.26, -0.51, -0.76\}$.

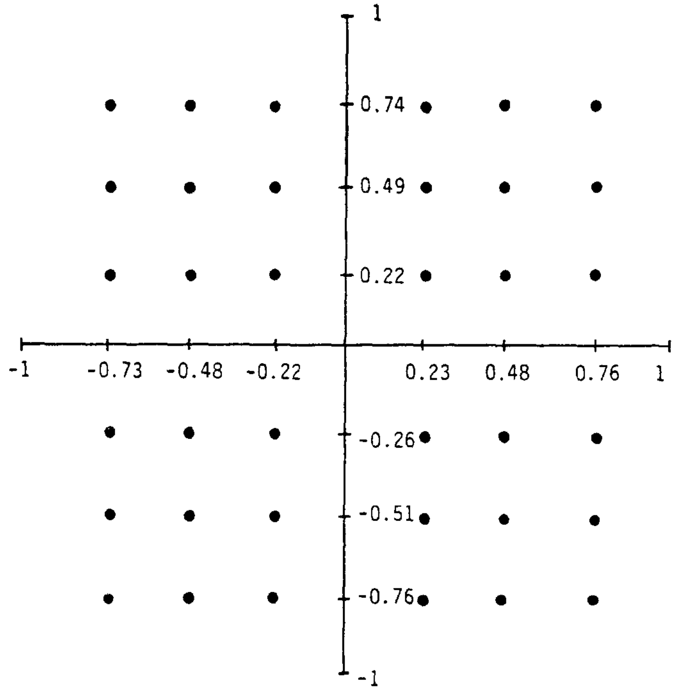


Fig. 5.1

The numerical results can be found in Table 5.1. Remark the fact that, because of the poles of the Beta function the polynomial approximation is inaccurate. The branched continued fraction turns out to be a far better approximation than Chisholm's Padé approximant.

Table 5.1

	(−0.75, −0.75)	(−0.50, −0.50)	(−0.25, −0.25)	(0.25,0.25)	(0.50,0.50)	(0.75,0.75)
Newton series	9.83	0.006	−6.785	7.423	3.1406	1.69449
Branched cont. fr.	9.884	0.0008	−6.7778	7.416301	3.14159245	1.69442617
Chisholm's P.A.	7.0	−0.14	−6.787	7.416310	3.14159269	1.69442617
$B(x,y)$	9.88839829	0.	−6.77770467	7.41629871	3.14159265	1.694426166

In class (B) we are going to compare the following approximations:

$$-\varphi[x_0][y_0]+\sum_{k=1}^{3}\frac{x-x_{k-1}|}{|\varphi[x_0,\ldots,x_k][y_0]}+\sum_{k=1}^{3}\frac{y-y_{k-1}|}{|\varphi[x_0][y_0,\ldots,y_k]}$$

$$+\sum_{j=1}^{2}\frac{(x-x_{j-1})(y-y_{j-1})|}{|\varphi[x_0,\ldots,x_j][y_0,\ldots,y_j]+\sum_{k=j+1}^{3}\frac{x-x_{k-1}|}{|\varphi[x_0,\ldots,x_k][y_0,\ldots,y_j]}+\sum_{k=j+1}^{3}\frac{y-y_{k-1}|}{|\varphi[x_0,\ldots,x_j][y_0,\ldots,y_k]}}$$

(15 data around each of the 6 points (u_i, v_j) given above).

– $P_S^{(7)}/Q_S^{(7)}$ recursively calculated via the determinantal formulas where the interpolationset S is described below (15 data around each of the 6 points (u_i, v_j) given above).

– $\varepsilon_{12}^{(0)}$ which needs as input $\varepsilon_0^{(0)}, \ldots, \varepsilon_0^{(12)}$ (91 data).

Since we are going to choose our data in exactly the same way for each of the points (u_i, v_j), we shall now only describe the situation for $(u_i, v_j) = (-0.75, -0.75)$; the other datasets simply result by translation.

For the branched continued fraction we use the sets

$$\{x_0, \ldots, x_3\} = \{-0.8, \ -0.76, \ -0.73, \ -0.7\}$$

and

$$\{y_0, \ldots, y_3\} = \{-0.79, \ -0.77, \ -0.74, \ -0.71\}.$$

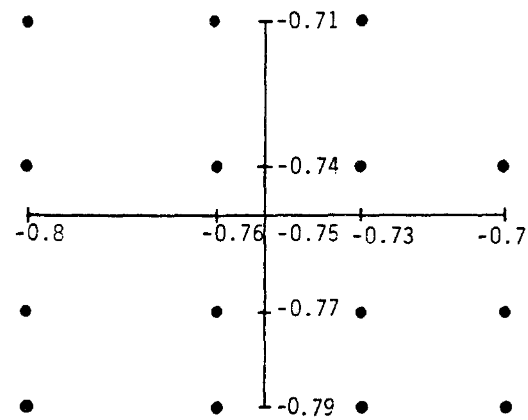


Fig. 5.2

For $P_S^{(7)}/Q_S^{(7)}$ we are given $S = \{(x_i, y_i) \mid i = 1, \ldots, 15\}$ drawn in the following figure:

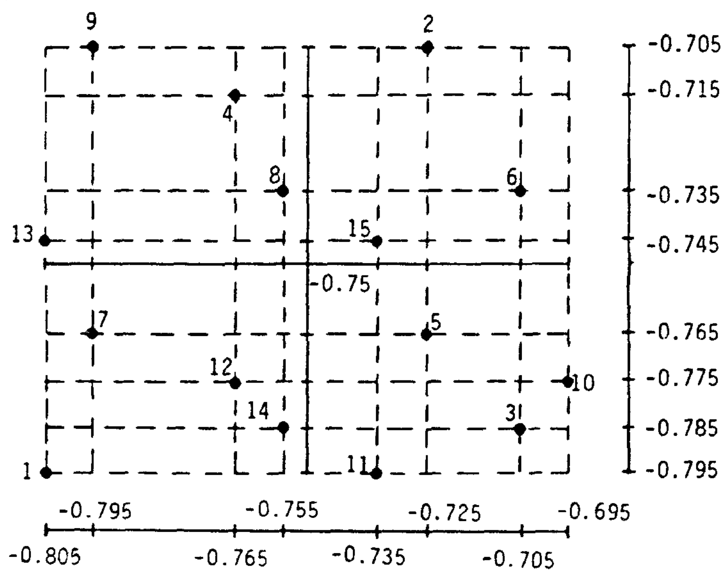


Fig. 5.3

In order to avoid the vanishing of certain determinants, the data are such that any three of the interpolationpoints never lie on a straight line; this simplifies the subdivision of S into smaller subsets. The numerical results are given in Table 5.2. The ε-algorithm and the branched continued fraction both give quite good results. What's more, they are also easier to program than the recursive method of section 3.

Table 5.2

	(−0.75, −0.75)	(−0.50, −0.50)	(−0.25, −0.25)	(0.25, 0.25)	(0.50, 0.50)	(0.75, 0.75)
Branched cont. fr.	9.8884	−0.000003	−6.777706	7.41629874	3.14159266	1.69442617
Recursive calc.	8.81	0.06	−6.787	7.4184	3.14152	1.694420
ε-algorithm	9.90	0.0005	−6.77767	7.41629871	3.14159265	1.69442617
$B(x,y)$	9.88839829	0.	−6.77770467	7.41629871	3.14159265	1.694426166

All the computations were performed in floating point double precision arithmetic with an input of 12 significant decimal digits. A last remark concerns the weight one should grant to the numerical results.

Small perturbations in the data do not very much affect the output to be found in Table 5.2, this is also true for the results in Table 5.1. So the numerical figures really represent the approximation power of the methods.

But we want to emphasize here that the rational interpolation methods depend very much on the numbering of the points. By renumbering the interpolation points one can obtain quite different results, more or less accurate.

References

[1] Chisholm, J.: Rational approximants defined from double power series. Math. Comp. *27*, 841 – 848 (1973).
[2] Cuyt, A.: The ε-algorithm and multivariate Padé approximants. Num. Math. *40*, 39 – 46 (1982).
[3] Cuyt, A., Verdonk, B.: General order Newton-Padé approximants for multivariate functions. Num. Math. *43*, 293 – 307 (1984).
[4] Graves-Morris, P., Hughes Jones, R., Makinson, G.: The calculation of some rational approximants in two variables. J. Inst. Math. Applics. *13*, 311 – 320 (1974).
[5] Hildebrand, F.: Introduction to Numerical Analysis. New York: McGraw-Hill 1956.
[6] Larkin, F. M.: Some techniques for rational interpolation. Computer J. *10*, 178 – 187 (1967).
[7] Murphy, J. A., O'Donohoe, M. R.: A two-variable generalization of the Stieltjes-type continued fraction. Journ. Comp. Appl. Math. *4*, 181 – 190 (1978).
[8] Siemaszko, W.: Thiele-type branched continued fractions for two-variable functions. J. Comp. Appl. Math. *9*, 137 – 153 (1983).
[9] Stoer, J.: Über zwei Algorithmen zur Interpolation mit rationalen Funktionen. Num. Math. *3*, 285 – 304 (1961).
[10] Thacher, H., Milne, W. E.: Interpolation in several variables. J. SIAM *8*, 33 – 42 (1960).
[11] Werner, H.: Remarks on Newton-type multivariate interpolation for subsets of grids. Computing *25*, 181 – 191 (1980).
[12] Wynn, P.: Über einen Interpolations-Algorithmus und gewisse andere Formeln, die in der Theorie der Interpolation durch rationale Funktionen bestehen. Num. Math. *2*, 151 – 182 (1960).

Annie A. M. Cuyt and
Brigitte M. Verdonk
Department of Mathematics and Computer Science
Universiteit Antwerpen (UIA)
Universiteitsplein 1
B-2610 Wilrijk
Belgium